

**Computer Applications and Innovative Teaching  
in  
Chemical Engineering Education**

A Thesis

submitted in fulfilment of the requirements for the Degree  
of  
Doctor of Philosophy in Chemical Engineering  
in the  
University of Canterbury  
by  
Judith Genista Mackenzie

University of Canterbury  
1998.

---

## DECLARATION

---

The work presented in this thesis is to the best of my knowledge and belief, original, except as acknowledged in the text. This material has not been submitted, either in whole or part, for a degree at this or any other University.

*Judith G. Mackenzie*

Judith Genista Mackenzie

January, 1998.

---

## ABSTRACT

---

In this thesis computer applications and innovative teaching methods in Chemical Engineering education were evaluated. These innovative teaching procedures and methods, incorporated into courses, were: co-operative learning in groups, creative problem-solving, a wider use of laboratory time, and very importantly computer technology. A paradigm shift from traditional teaching methods to reduce lectures, include different teaching methods and use the technology, was evaluated.

The literature review details developments in the design of the human computer interface. What constitutes a good interface design? The formation of relevant criteria for an effective interface design method is a complex problem. Human Computer Interface (HCI) research focuses on the design of the interface considering the interaction of people and technology. The new challenge for designers and educators is the implementation of learner-centred design.

Mathematical power tools - Maple<sup>TM</sup>, Mathematica<sup>TM</sup>, MATLAB<sup>TM</sup>, and Excel<sup>TM</sup> - are computer applications for simple and advanced mathematics, enabling engineers to solve mathematical and computational problems. These four packages were compared, assessing ease of learning and flexibility in use, for undergraduate engineers, using engineering problems as examples. Six problems were solved for comparison:- steady-state heat conduction temperature profiles were calculated and graphically presented, an energy analysis requiring inversion of an eighty by eighty matrix of input/output data, a multi-variable optimisation for an illumination design, a Fourier series approximation, and two process control problems; one root finding for Bode design and the other requiring integration of a system of differential equations.. The best mathematical application depended on the nature of the problem to be solved but overall Mathematica<sup>TM</sup> and MATLAB<sup>TM</sup> were ranked ahead of Maple<sup>TM</sup> and Excel<sup>TM</sup>.

A telephone survey of Christchurch Engineers was conducted to find out which mathematical packages were used by consulting engineers. Results indicated that 64% of those surveyed used Excel, 4% used Mathematica, 4% used MATLAB and 28% did not use any of the mathematical applications.

The development of problem solving skills is an integrated part of the teaching of design in third year. Computer-based problem solving modules were used as a supplement to lectures. Students worked in pairs in a computer laboratory situation, with their work assessed on-line. Students' opinions of the problem solving modules were evaluated by a questionnaire at the end of the course. The interactive computer modules provided a new and different learning environment which the students found to be a useful supplement to lectures.

Problem solving skills, at least for the small student population used in this work, were found to be independent of academic achievement. Strategies for problem

solving are techniques that can be learned. In addition, the research examined four further questions in relation to problem solving skills: how students performed on each computer module, gender differences of our Chemical Engineering students in problem solving skills, the relationship between problem solving skills and ability in other subjects, and attempted to assess the academic ability of Chemical Engineering students from year to year.

A computer-based simulation exercise - the Amoco project - was used as the final assignment for the third year chemical engineering design class. This project provided an industrial based example for planning, decision making and problem solving. A questionnaire administered at the end of the course gave students' responses to the course. Working in groups of three was a feature of the project and the majority of students enjoyed this aspect, as well as finding the computer simulation a useful learning experience. Pre and post-test results showed a significant learning outcome from this project in 1997.

Production of a multimedia interactive computer module for teaching finite difference approximations to first and second derivatives was the final project. This involved analysis, design, development, implementation and evaluation. To evaluate the computer module a pilot study of eighteen voluntary students, and an experimental group of eighty students were used. Supplementary to lectures, two different laboratory sessions compared the computer-based module with a mathematical problem solving session. In the pilot study there was a significant difference between the groups ( $t=3.81$ ), but there was no significant difference between the two methods for the experimental group ( $t=1.69$ ). An Analysis of Variance resulted in an F-test value of 2.29 which was also not significant.

Learning Styles were assessed by Solomon's Learning Styles Inventory indicating that learning styles were different from those addressed by traditional lectures. Students have different learning styles, and therefore educators need to provide a variety of teaching and learning methods.

The innovative contribution of this work to the field of Chemical Engineering education included:

- Comparison of four mathematical packages using six engineering problems to determine the suitability of these packages for undergraduate students; evaluation of computer-based problem solving modules as a supplement to lectures;
- Setting up and maintaining the Amoco design project on our Unix system for students use on a PC display network;
- Adapting the Amoco project to suit third year engineering students and measuring the learning outcome;
- Designing and evaluating a computer-based module on finite difference approximation for a laboratory setting as a supplement to lectures;
- Comparison of two different teaching methods - text and computer-based to supplement lectures on numerical methods;
- Evaluation of a selection of engineering students learning styles.



---

## ACKNOWLEDGMENTS

---

I would like to acknowledge the support and guidance of my supervisor, Associate Professor Brian Earl, who has helped and encouraged me during the past year. In addition, I must thank Associate Professor Maurice Allen who, prior to his appointment at Murdoch University, Perth, Western Australia, was my supervisor, and co-author of several papers. Dr Allen's support and financial assistance to attend conferences were greatly appreciated.

This thesis would not have been possible without the assistance and co-operation of the staff and students of the Department of Chemical and Process Engineering to whom I express my appreciation. Also, my thanks to the Postgraduate students for their advice and friendship. A Departmental Scholarship in 1996 and a grant-in-aid to attend the CHEMECA'97 Conference were greatly appreciated.

Finally, and most importantly, I would like to thank my husband Don, and our four children, Anne, Nicola, Thomas and Susan for their patience and encouragement during the last few years.

---

## CONTENTS

---

	Page
Declaration.....	iii
Abstract.....	iv
Acknowledgements.....	vi
Contents .....	vii
Illustrations .....	x
Tables.....	xi

### Chapter 1 Introduction

1.1 Computers as a Tool, Tutor and Simulator.....	1-1
1.2 Computing in Chemical Engineering Education .....	1-1
1.3 Styles of Learning and Teaching .....	1-2
1.3.1 Experiential Learning Theory .....	1-2
1.3.2 The 4MAT Model .....	1-2
1.3.3 Learning Styles of Engineering Students.....	1-3
1.4 Objective of this Research .....	1-4
1.5 Scope of this Research .....	1-4
1.6 An Overview of this Thesis .....	1-5

### Chapter 2 Human Computer Interaction - Literature Review

2.1 Human Computer Interaction.....	2-1
2.2 Craft Method of Interface Design .....	2-2
2.3 Cognitive Engineering .....	2-3
2.3.1 Cognitive Metric Methods .....	2-4
2.3.2 Grammar Models.....	2-4
2.3.3 Knowledge Models .....	2-4
2.3.4 User Modelling Methods.....	2-4
2.4 Enhanced Software Engineering.....	2-4
2.5 Technologists Approach .....	2-4
2.6 Computer Interface Design? .....	2-4
2.7 Learner Centred Design .....	2-4

### Chapter 3 Mathematical Power Tools

3.1 Introduction .....	3-1
3.2 Mathematical Tools.....	3-2
3.3 Engineering Problem Solving .....	3-2
3.3.1 Two-Dimensional Heat Transfer.....	3-3
3.3.2 Energy Analysis .....	3-4
3.3.3 An Illumination Design by Optimisation.....	3-5
3.3.4 Fourier Series Approximation.....	3-6
3.3.5 Root-finding for Control System Design.....	3-8
3.3.6 Differential Equation Solution for Time Response.....	3-10
3.4 Evaluation .....	3-12
3.5 Engineers and Mathematics .....	3-13

### Chapter 4 Creative Problem Solving Skills

4.1 Problem Solving.....	4-1
4.2 A New Approach to Engineering Design.....	4-1
4.3 Teaching Aims and Objectives.....	4-2
4.4 Research Aims and Objectives.....	4-2
4.5 Creative Problem Solving Modules .....	4-3
4.5.1 Introduction to Problem Solving.....	4-3
4.5.2 Problem Definition Techniques .....	4-3
4.5.3 Brainstorming-Solution Generation.....	4-3
4.5.4 Situation Analysis .....	4-4
4.5.5 Problem Analysis-Finding the True Cause .....	4-5
4.5.6 Decision Analysis-Choosing the Best Solution .....	4-4
4.5.7 Potential Problem Analysis - Avoiding Future Problems .....	4-4
4.5.8 Planning-Implementation of Solutions .....	4-5
4.5.9 Evaluation-Solution Evaluation Techniques.....	4-5
4.6 Course Organisation, Assessment and Evaluation .....	4-5
4.7 Results and Discussion.....	4-5
4.8 Further Research Questions.....	4-9
4.8.1 Assessment on Each Module.....	4-9
4.8.2 Gender Difference .....	4-10
4.8.3 Relationship With Other Subjects .....	4-10
4.8.4 Academic Ability-Bursary Marks .....	4-11

	Page
<b>Chapter 5 Amoco Computer Simulation</b>	
5.1 The Amoco Project .....	5-1
5.2 Teaching Aims and Objectives .....	5-1
5.3 Research Aims and Objectives .....	5-2
5.4 Amoco Resid Hydrotreater .....	5-2
5.5 The Computer Simulation.....	5-3
5.5.1 Use of the Amoco Simulation .....	5-3
5.5.2 Budgetary Constraints.....	5-4
5.5.3 Planning Strategy .....	5-5
5.5.4 The Interim Report .....	5-6
5.5.5 Control Strategy .....	5-6
5.6 Evaluation .....	5-7
5.7 Results and Discussion .....	5-7
5.8 Learning Outcomes .....	5-9
 <b>Chapter 6 Finite Difference Approximations</b>	
6.1 Computer Aided Learning .....	6-1
6.2 Research Aims and Objectives .....	6-1
6.3 Production of the FDA Computer Module .....	6-2
6.3.1 Prototype and Development Process .....	6-2
6.3.2 Disadvantages of the Traditional Development Methodology .....	6-3
6.3.3 A New Development Approach.....	6-3
6.4 Pilot Study .....	6-5
6.5 Experimental Study .....	6-6
6.5.1 Handout Versus Module .....	6-6
6.6 Soloman's Learning Styles .....	6-8
 <b>Chapter 7 Conclusions</b>	
7.1 Evaluation .....	7-2
7.2 Learning Skills.....	7-3
7.3 Recommendations For Further Research.....	7-3
 <b>Chapter 8 References</b>	
 <b>Appendices</b>	
Appendix 1 - List of Publications.....	A-1
Appendix 2 - Finite Difference Approximations - Handout.....	A-2
Appendix 3 - Finite Difference Approximations (CD ROM).....	A-3

---

## ILLUSTRATIONS

---

	Page
Figure 3.1 : Contour plot of plate temperature distribution from Excel.....	3-3
Figure 3.2 : Plate temperature distribution .....	3-4
Figure 3.3 : A 3D contour plot shows the illumination of the bicycle pad .....	3-6
Figure 3.4 : Graphical subplots of the input function using Fourier series .....	3-7
Figure 3.5 : Control System Block Diagram.....	3-9
Figure 3.6 : Mathematica output to confirm location of angular frequency root ....	3-9
Figure 3.7 : Time response in Mathematica - set point response .....	3-10
Figure 3.8 : Control system initial condition response.....	3-11
Figure 3.9 : Excel graph of time response to a unit step in the load, u .....	3-12
Figure 3.10 : Mathematical tools used by Christchurch engineers .....	3-13
Figure 4.1 : Marks for assignment 1 compared with adjusted GPA .....	4-6
Figure 4.2 : Assignment 2 mark compared with adjusted GPA .....	4-6
Figure 4.3 : Students support the interactive computer modules as a useful supplement to lectures.....	4-7
Figure 4.4 : Students' opinion of the value of the interactive problem solving modules.....	4-7
Figure 4.5 : Students' response to computer-based assessment.....	4-8
Figure 4.6 : Students' response to working in pairs.....	4-8
Figure 5.1 : Amoco hydrodesulphurization pilot plant .....	5-3
Figure 5.2 : Students' marks for the design project in 1996 and 1997 .....	5-7
Figure 5.3 : Group project marks compared with adjusted GPA (1996).....	5-7
Figure 5.4 : Students' response to computer simulation as a learning experience ..	5-8
Figure 5.5 : Students' response to working as a member of a team .....	5-8
Figure 5.6 : Mean scores on the Amoco project pre and post tests .....	5-9
Figure 6.1 : The traditional five step development process.....	6-2
Figure 6.2 : Iterative phases of multimedia development .....	6-4
Figure 6.3 : Comparison of pre-test and post-test mean scores.....	6-7

---

## TABLES

---

	Page
Table 1.1 : Five dimensions of learning styles.....	1-3
Table 2.1 : Approaches to interface design.....	2-2
Table 3.1 : Evaluation of mathematical problem solving tools .....	3-12
Table 4.1 : Students' comments on Wimbey's method of problem solving .....	4-6
Table 4.2 : Students' assessment on each computer module in 1996.....	4-9
Table 4.3 : Students' assessment on each computer module in 1997.....	4-9
Table 4.4 : Comparison of male and female students' marks for problem solving	4-10
Table 4.5 : Course codes for second year Chemical Engineering students.....	4-10
Table 4.6 : Correlation of subject and problem solving marks .....	4-10
Table 4.7 : Bursary marks for 1996 and 1997 third year Chemical Engineering students .....	4-11
Table 5.1 : Amoco project timetable.....	5-4
Table 5.2 : Expenses for Amoco project.....	5-5
Table 5.3 : Advantages and disadvantages of working in groups.....	5-9
Table 5.4 : Students' comments about the Amoco project.....	5-10
Table 6.1 : Results of pilot study pre-test .....	6-5
Table 6.2 : Results of pilot study scores .....	6-6
Table 6.3 : Mean GPA for the two groups - Handout versus Module .....	6-6
Table 6.4 : Results of the differences in pre and post test scores.....	6-7
Table 6.5 : Results of Learning Styles survey .....	6-8

### **1.1 COMPUTERS AS A TOOL, TUTOR AND SIMULATOR.**

Computer-based instruction has the potential to stimulate teaching and learning. Engineering courses can be enhanced by computer simulation and computer-assisted instruction. Seader, (1989) comments:

*The microcomputer is the most powerful learning device since the printed textbook. If the computer is used properly, individual interactive learning is economically possible for the first time.*

Computers in education can be used as a tool, a tutor or as a simulator. All three pedagogues have been studied as part of this research;

- as a *tool* using Mathematical Packages.
- as a *tutor* in Strategies for Creative Problem Solving.
- as a *simulator* in the Amoco Project.

To be effective as a learning device, the design of the computer interface is an important consideration, and one of the reasons for a rapidly developing field of research. What constitutes a good interface design? The criteria for an effective interface design methodology is a complex problem. HCI research focuses on the design of the interface considering the interaction of people and technology. The new challenge for designers and educators is the implementation of learner-centred design.

### **1.2 COMPUTING IN CHEMICAL ENGINEERING EDUCATION**

In an Award Lecture on Computing in Engineering Education (Carnahan,1992) discussed the developments in academic computing that had occurred in the curriculum since the early 1960's in the USA. Carnahan comments that:

*...the creation of good instructional software is very demanding of faculty time, student manpower, and money. ...And since the reward structure, at least at research universities, discourages such activities, there is little prospect of major faculty efforts to develop instructional software except in association with popular textbooks or when supported by outside funding.*

Carnahan, Fogler and others at the University of Michigan have worked on instructional modules for personal computers for many years. More recently, Fogler received a National Science Foundation (NSF) curriculum improvement grant which enabled the development of instructional modules for chemical engineering

thermodynamics, reaction engineering, separations, fluid mechanics and problem solving.

Two other NSF-sponsored curriculum improvement grants were mentioned in the Award Lecture. One project (co-directed by Reklaitis and Squires at Purdue University), enabled the development of comprehensive simulation models of industrial processes. The simulations, including professionally produced videos, were designed for use in a simulation laboratory, design course, or engineering-science courses.

The second project (directed by B. Finlayson at the University of Washington) focused on numerical intensive solution of general fluid mechanics, reaction, and heat/mass transfer models: its major features were extensive graphical input and high-quality colour display for visualization of computer results.

Case studies, for example, Young and Svrcek (1996) provide models to demonstrate how industry benefits from computer simulation. Studies of the controllability and operability of steady state and dynamic simulation for process response and process control were undertaken at the design stage before plant construction; the responses subsequently being used for development of the final control scheme.

### **1.3 STYLES OF LEARNING AND TEACHING**

Students learn in many different ways - by seeing, hearing; reflecting and acting; reasoning logically and intuitively; memorizing and visualising; steadily and in fits and starts. Teaching methods vary and many of the different learning styles are not always addressed by the traditional lecturing style of University lectures (Laurillard, 1996).

#### **1.3.1 Experiential Learning Theory**

Kolb's Experiential learning theory (1984) is grounded in the idea that individuals test ideas on actual experiences. According to Kolb, human learning and personal development are synonymous processes which involve the continuous integration of a distinct set of independent systems that give meaning to life's circumstances.

At the heart of Kolb's theory is the conviction that learning is a continually recurring process through which individuals refine and integrate basic adaptive modes for perceiving, thinking, acting and feeling.

#### **1.3.2 The 4MAT Model.**

The 4MAT model (right /left brain mode) (McCarthy, 1987), maintains that effective instruction embraces four essential phases of learning:



- Learner motivation
- Conceptual mastery
- Application of ideas
- Creative synthesis

The basic premise of 4MAT is that while students favour different learning places on the 4MAT cycle according to their learning styles, they all need to go through the four major steps when learning anything. McCarthy believes that successful learning begins through the creation of personal meaning for students, proceeds to conceptual understanding, then to application and finally integration. The learning styles of students results in varying levels of learner comfort as they move through the cycle. Students experience their most comfortable place, while being stretched to learn in ways that are more challenging for them.

### 1.3.3 Learning Styles of Engineering Students.

A model of learning styles, intended for engineering education has been outlined by Felder and Silverman, (1988); shown in Table 1.1.

**Table 1.1 : Five Dimensions of Learning Styles.**

<u>Sensory</u> / <b>Intuitive</b>	Sensors prefer facts, data and experimentation, are careful and patient with detail, but may be slow. Intuitors prefer concepts, principles, and theories, and may be quick but careless.
<u>Visual</u> / <b>Verbal</b>	Visual learners prefer pictures, diagrams, charts, movies, demonstrations, and exhibitions. Verbal learners prefer words, discussions, explanations, formulas and equations.
<u>Inductive</u> / <b>Deductive</b>	Inductive learning develops principles and generalities from observations, the natural human learning approach. Deductive development starts with governing principles and then develops applications, the natural teaching approach.
Active / Reflective	Active learners learn by doing and participating. Reflective learners learn by thinking or pondering introspectively. Unfortunately, most lectures provide opportunity for neither approach (passive)
<b>Sequential</b> / Global	Sequential learners take things step by step, and will be partially effective with partial understanding. Global learners must see the whole picture for any of it to make sense, and are completely ineffective until they suddenly understand the entire subject.
<u>Underlined</u>	Preferred by most engineering students
<b>Bold Type</b>	Preferred by most engineering lecturers

Learning styles of most engineering students and teaching styles of most engineering lecturers are often incompatible. Felder and Silverman's learning styles (Table 1.1) provide a guide for the development of instructional computer-based teaching modules and simulations to effectively improve teaching and learning. Solomon (1992), together with Felder, has developed an Inventory of Learning Styles to identify different learning styles.

#### **1.4 Objective of this Research.**

The objective of this thesis is: *To assess computer applications and innovative teaching methods in Chemical Engineering education.*

The key attributes and desirable features of the methodology are:

- evaluation
- learning styles
- keep it simple
- learning-centred design of the user interface.
- significant learning outcomes
- enthusiastic and contented students.

#### **1.5 Scope of this Research.**

The starting point for this research was a review of the literature on the design of the human computer interface, with the latest development being learner-centred design. Educational research has shown the benefits to teachers and students of using different teaching procedures and methods to provide an optimum learning environment. However, many of these alternative teaching practices are not incorporated in higher education degree courses where traditional methods of lecturing and assessment continue. Innovative teaching procedures and methods, as part of this research, have been partly incorporated into some Chemical Engineering courses. Some of these are: co-operative learning in groups, problem-based learning, a wider use of laboratory experiments, and very importantly computer technology.

Why institute changes in traditional teaching practices? While moving to markedly poorer staff/student ratios, efficient changes may be necessary to maintain educational standards. The search for excellence in teaching and learning is the goal of teachers and students alike. A paradigm shift from traditional teaching methods to reduce lectures, include different teaching methods and use the technology was evaluated in this thesis.

## 1.6 An Overview of this Thesis.

Following a review of human computer interaction and approaches to interface design in Chapter 2, this thesis evaluates computer applications and innovative teaching in Chemical Engineering education.

- A comparison of four mathematical power tools<sup>1</sup> - Maple™, Mathematica™, MATLAB™ and Excel™, is made using six engineering examples (Chapter 3).
- A problem solving approach has been adopted for teaching engineering design to third year students. The interactive computer instruction package *Strategies for Creative Problem Solving* was evaluated (Chapter 4).
- The *Amoco Project*, a computer-based simulation exercise, was used as the final assignment for the third year design class. This project provided an industrial based example for planning, decision making and problem solving. Students response to this computer simulated learning experience was evaluated. (Chapter 5).
- The design and evaluation of computer-based learning of Finite Difference Approximations was compared with text-based learning of the same material as a supplement to lectures (Chapter 6).
- Learning styles of Engineering students were assessed using Solomon's Inventory of Learning Styles (Chapter 6).

---

<sup>1</sup> Whether upper or lower case the words Maple, Mathematica, MATLAB and Excel are registered trademarks for the equivalent mathematical computer packages with these names.



## **LITERATURE REVIEW**

### **2.1 HUMAN COMPUTER INTERACTION**

This literature review covers a Human Computer Interaction approach to the design of the computer interface. Human computer interaction (or HCI) is a rapidly developing field of research which considers problems related to the *usability* of a system. Current HCI research emphasises:

- the user should be the central focus of the system design;
- interfaces that are both easy to learn and to use are essential;
- the user should control the system not the machine controlling the user;
- multidisciplinary study with contributions ranging from psychology to graphic design.

One of the main points of debate amongst HCI researchers appears to be the way interface design is approached. What are the features of good interface design? Should the design process be craft-orientated, an applied science or an engineering discipline? This occurs because the whole HCI is very multidisciplinary. Whichever path is chosen sends the researcher off in one of many directions of this complex field.

*Any system which cannot be well taught to a layman in ten minutes, by a tutor in the presence of a responding set-up, is too complicated.*

(Nelson, 1974)

Four major approaches to interface design identified by Wallace and Anderson (1993) are as follows:

- craft
- cognitive engineering
- enhanced software engineering
- technologist

First, the craft approach where every design project is viewed as unique, evolves under the guidance of a skilled human factors expert, depending on the circumstances. Second, cognitive engineering is the most theoretical approach to interface design. Cognitive psychology has been applied to the problems facing designers to facilitate optimum design. Third, the enhanced software engineering approach has made attempts to introduce HCI techniques into the field of traditional systems engineering. Finally, the technologist approach has had more and more impact in recent years, as this approach tries to provide software tools to solve the problems of interface design. Each of these approaches is briefly summarised in Table 2.1.

**Table 2.1:** Approaches to Interface Design

	Craft Approach	Cognitive Engineering	Enhanced Software Engineering	Technology
Researchers	Laurel	Norman Carroll	Sutcliffe	Wasserman Hix and Hartson
Philosophy	Craft-orientated design through skill and experience	Apply the psychological knowledge base to achieve optimal design	Incorporate HCI into software engineering	Quantify and automate the design process
Character	Evolutionary	Psychologists Ergonomist Human factors specialists	Structured transformation	Black box generation
Focus	Specification Design	Specification Evaluation	Specification	Implementation
Tools	Brainstorming Prototyping User evaluation	Task analysis methods GOMS, CCT, KAT	Software engineering methods CASE tools	UIMS UAN

Each of the four approaches will be looked at in detail, analysing their strengths and weaknesses, their design philosophies, and identifying some of the major methods within each tradition. What constitutes quality in interface design is the chief question to be answered.

## 2.2 CRAFT METHOD OF INTERFACE DESIGN

Solutions evolve to meet specific needs. 'Craft' implies a process in which there is little or no separation between design and manufacture.

“ Building human-computer interfaces involves applying the relevant knowledge in a complex problem solving context to systems of tasks and artefacts too complex to be completely understood. In practice, the distinction between design and implementation are necessarily so blurred that the construction of the human computer interface can surely be considered a craft”.

(Wroblewski, 1991)

As each design is unique, successful design can only emerge through experimentation and prototyping. There is no need for a detailed understanding of the users' cognitive process as the users themselves will participate fully in the development of the system. Supporters of this approach tend to believe that a structured methodology for the interface design is impossible as the aesthetics of interface design do not lend themselves to analytical techniques. The designer in this context is regarded as a craft person and in some cases an artist.

Historical accident has kept programmers in control of a field in which most of them have no aptitude: the artistic integration of the mechanisms they work with..... Learning to program has no more to do with designing interactive software than learning to touch type has to do with writing poetry.

Nelson (1990).

From the craft viewpoint, what interface designers need is talent, not methodology. Much interface design has been achieved through craft (Oren, 1990, Smith, 1982).

Yet while it may not be rigorous or scientific, the craft method does work. The ground breaking interface design of the Apple Mac is an example of what can be achieved by the craft approach. However, to suggest that this is the only way to develop effective human - computer interfaces, or even the best way, is to overlook important problems. Crafts people learn by doing, they have no specific knowledge of the principles underlying successful design.

## **2.3 COGNITIVE ENGINEERING**

In the early 1980's many cognitive psychologists began to enter the field of HCI. They applied the theories of information processing and problem solving developed by earlier psychologists, to the problem of interface design.

It appeared to the psychologists that when users interacted with computers they were interacting with information. Consequently, the field of HCI seemed ideal as: "...an arena in which potential understanding of human mental powers and limitations can be tested.

Lander (1987).

Initially, HCI researchers attempted to develop extensions of psychology. Many techniques and methods emerged which can be divided into four broad categories: cognitive metric methods, grammar models, knowledge models and user modelling methods.

Of the various cognitive models, the most important was the model human processor (Card et al, 1983) which described a psychological model of humans consisting of three interacting systems; the perceptual, motor and cognitive systems, each of which has its own memory (that is internal representation or knowledge) and processor.

This model led to the GOMS (Goals, Operations, Methods and Selection) rules and CCT (Cognitive Complexity Theory) methods of task analysis.

**2.3.1 Cognitive Metric Methods** - are keystroke level and unit-task-level methods (Card and Moran, 1983) which set out to measure the users' performance time and memory loading for basic text editing tasks. The aim was to provide computed records of human performance which could be traded against other engineering variables. The main characteristic of these methods was to try and quantify the features of users so they could be considered during systems design.

**2.3.2 Grammar Models** (Simon, 1988) - based on the assumption that users develop mental models of the tasks they perform. The concept of mental models in psychological theorising and HCI research has developed in many ways. A definition by Norman (1988) is

*...the model people have of themselves, others, the environment, and the things with which they interact. People form mental models through experience, training and instruction.*

There are several other cognitive task analysis techniques which focus on different aspects of the general information processing model. Task Action Grammar (TAG; Payne and Green, 1989) is concerned with an evaluation of the learnability of systems. Moran's Command Language Grammar (CLG; Moran, 1981) is a form of grammar comprising a symbolic notation consisting of a sequence of hierarchical levels, each being a refinement of the preceding level.

**2.3.3 Knowledge Models** - Johnson's theory of Task Knowledge Structures (TKS; Johnson, 1992) assumes that as people learn and perform tasks, they develop knowledge structures. A method known as Knowledge Analysis of Tasks (KAT) is utilised "... to identify the elements of knowledge represented in a task knowledge structure" (Johnson, 1992).

**2.3.4 User Modelling Methods** - are methods which aim to develop models of cognition. These methods describe not just what a user must know to perform a task, but how that knowledge is obtained and used during the execution of the task. The goal is "...a quantitative understanding of what might be going on in a users' head rather than a purely quantitative understanding of how long the average head is going to be busy". (Barnard, 1991). These methods try to incorporate their cognitive models in the software tools, which can make the theory accessible to software developers. Examples of this approach include Programmable User Models (PMU; Young et al, 1989) and Approximate Modelling of Cognitive Activity (Barnard et al, 1988).



## 2.4 ENHANCED SOFTWARE ENGINEERING

User software engineering is a methodology, supported by automated tools, for the systematic development of information systems. In the early stages of the software development process, the user software development method gives particular attention to effective user involvement, concentrating on external design and the use of quickly created and modified prototypes of the user interface. The user software engineering methodology is supported by an integrated set of graphically based tools. Seven goals of this methodology are described by Wassermann (1987) as functionality, reliability, usability, evolvability, automated support, improved developer productivity, and reusability. Aspects of the requirements analysis include data modelling, activity modelling, analysis of user characteristics, and analysis of usage characteristics. Rather than proceed with further refinements of the system functions, following a traditional “top-down” approach, the user system engineering methodology follows an “outside in” approach, in which the external interface to the system is defined. However, they still rely on traditional software methods for much of their design input and consequently do not address the problem of interface design.

An object-orientated task analysis method called ATOM (Walsh, 1989) has been developed to integrate with the JSD method (Jackson, 1983). Both of these methods aim to include HCI throughout the software development life-cycle.

Another approach to overcome the problem of interface design is to try and extend software engineering methods by introducing task analysis into the life cycle (Damodaran et al, 1988, Sutcliffe, 1988). By supplementing the requirements analysis stage with a task analysis technique, the software engineering method can better identify the needs of users and provide information for the specification of the user interface. “It is important to make the introduction of an HCI component attractive to analysis, if it is to gain popularity and subsequently impact on systems design” Wallace and Anderson, (1993).

The provision of Computer-Aided Software Engineering (CASE) tool support can do a lot to assist the commercial adoption of HCI methods. CASE tools can address many of the drawbacks associated with HCI methodologies (Carter, 1991; Macaulay, 1993). However, while HCI methods remain open to criticism, that they are too difficult to learn and not useful enough to justify the effort, CASE tools by themselves will not be sufficient to sell these methods to developers. “Attempting to improve interface design by enhancing existing software engineering methods is by far the most pragmatic stand of HCI research” (Wallace and Anderson, 1993).

## 2.5 TECHNOLOGISTS APPROACH

Many researchers in HCI look to technology for the answer to the problem of poor interface design. Computer systems are often sold to the user on the strength of their functions. Consequently, once the functions have been designed, developers find

themselves under considerable commercial pressure to release the system. As a result of these pressures, the interface is unlikely to get the design effort it requires.

The answer to this problem, from the technologists point of view is to provide automated development tools that will free programmers from mundane and time consuming tasks and leave them more time to spend on design.

*The objective is to free the application programmer from low-level details so as to be able to concentrate on higher application-specific aspects of the user interface.*

Buxton et al (1983).

Technologists also encourage rapid prototyping as a means of requirements capture for interface design. Automated tools which allow rapid development of complex systems have certainly made prototyping a more attractive proposition for software developers.

An important concept in the world of computer-aided software development is the User Interface Management System (UIMS). This is a major focus of interest for technologists.

UIMS consists of three parts:

- A design environment, concerned with the development of screen elements such as icons and menus plus the system dialogue specification.
- A linkage module, which defines the operational characteristics of a dialogue and links the interface components to the underlying functions of the application.
- A management function, that controls interaction during run-time.

Early attempts at the UIMS such as Menulay (Buxton et al, 1983), Syngraph (Olsen and Dempsey, 1983) and MacApp (Schmucker, 1986) provided limited facilities and appeal. However, more sophisticated products have recently been developed which have greater potential for improving the interface development process: products such as UIM/X (Visual Edge), DataViews (V.I.Corp), SuperCard (Aldus) and Dialog (Hewlett Packard)

A notation created at Virginia Tech, (Hartson et al, 1990; Hix and Hartson, 1989, 1993) the User Action Notation (UAN) is a notation to write down the actions that the user will perform with the system and system responses, from the users point of view. A second topic Hix and Hartson (1993) are working on is a tool to help with a usability evaluation process. The tool is called IDEAL, which stands for Interface Design Environment and Analysis Lattice, and supports all activities in the usability evaluation process. It provides connections between, for example, user task descriptions written in UAN, the appropriate benchmark tasks for users to perform during an evaluation session, the usability goals for each task, and then capturing and analysing the results of observing users performing their tasks.

As well as developing the product, Hix and Hartson (1993) emphasise the importance of the process (evaluation) when designing the interface. User opinion measured subjectively, is a strong factor in user satisfaction, which may actually affect user performance over a longer period of time.

Subjective and objective measurements which can be used for evaluating the design of the interface are as follows:

1. Benchmark task - How the user should do the task.

Measuring a user's performance on a benchmark task provides an objective usability metric for the related usability attribute.

2. Time to complete a task
3. Number of errors during task performance.

A questionnaire can be used to determine a users' subjective satisfaction with the interface; measuring a users' satisfaction provides a subjective (but still quantitative) usability metric for the related usability attribute.

The Questionnaire for User Interface Satisfaction (QUIS) developed at the University of Maryland (Chin et al, 1988), is an extensive and thoroughly validated questionnaire for determining subjective interface usability considering:

- screen
- terminology and system information
- learning
- system capabilities

Values to be measured include:

- Time to Complete task
- Number of percentage of errors
- Percentage of tasks completed in a given time
- Ratio of successes to failures
- Time spent in errors and recovery
- Number of commands/actions used to perform tasks
- Frequency of help and documentation use
- Number of repetitions of failed commands
- Number of available commands not invoked
- Number of times user expresses frustration or satisfaction

However, Whiteside et al (1988) state that there are cautions to be considered in developing and using usability specifications, criticising them for being too scientific in nature which may present a distorted impression of the product's real usability in the workplace.

An important feature of each of the four approaches to interface design: craft, cognitive engineering, software engineering and technologists, is their perception of quality in interface design.

## **2.6 COMPUTER INTERFACE DESIGN**

What constitutes an effective computer interface design? Detailed guidelines and recommendations for design of usable systems for HCI are outlined by Shneiderman (1987), Tognazzini (1989), Carroll et al (1987). Apple's Human Interface Guidelines (Apple, 1987) claims that ten fundamental principles guide the design of the Apple Desktop Interface. The principles are quite general, although they are elaborated with examples and discussed informally. Apple applications are often cited as the model example of interface design, developed over the years by the efforts of the interface evangelist 'Tog' (Tognazzini, 1989), and the team at Apple Computers Inc.

Norman (1988) in his book on the design of everyday things outlines four principles of good design:

- Visibility
- A good conceptual model
- Good mapping
- Feedback

An example of bad design is the ticketing arrangements for the Monorail in Sydney. It is so poorly designed that an attendant is needed to explain the system to the users.

Design requires creative skills and knowledge for the production of everyday things. In considering the design of the interface it is well to remember the words of Winogard (1993), "I think the challenge is to really keep the knowledge of both the technology and the people playing off against each other in order to develop new things".

## **2.7 LEARNER CENTRED DESIGN**

Interface design efforts have traditionally focused on the needs of the user - a person with a familiar task at hand who accesses the computer to help facilitate the completion of that task. As technology continues to transform the traditional classroom environment with computer -based training tools, software designers are faced with a new dilemma; how to create systems for users who may not understand the task to be performed, let alone the software to support that effort.

Today, the great challenge for designers of educational tools and technologies is to create interfaces that allow learners to grasp the elements of the assignment as well as the software. This emerging area is called Learner-Centred Design (LCD).

Norman and Spohrer (1996) describe the approach to LCD as :

*The basic issues can be described through such keywords as "constructivism", "learner-centred", and "problem-based". At the heart of the idea is that people learn best when engrossed in the topic, motivated to seek out new knowledge and skills because they need them to solve the problem at hand. The goal is active exploration, construction and learning rather than the passivity of lecture attendance and textbook reading.*

Learner Centred Design offers challenges to designers, educators and everyone engaged with the problem of learning.



### **3.1 INTRODUCTION**

The concept of using computers for helping to solve mathematical problems has particular appeal to engineers. Prior to the personal computer, programs were coded in a computer language, often FORTRAN, for a very specific application. Later, generalised packages were developed to solve problems in particular disciplines. Some were purely calculational, to solve differential equations or to invert matrices; others had more of an engineering flavour. Such specialised engineering software was powerful but with a narrow focus; for example, programs for electrical circuit design, civil engineering structural calculations, or flowsheeting programs for the design of chemical processing plants.

Software programs with a strong emphasis on calculation and numerical evaluation continue to be marketed for the personal computer. Reviews of individual packages outline the latest features, for example, Maple V, Release 3 (Hutton and Hutton, 1995, Foster, 1996), Mathematica 3.0 (Studt, 1996, Vaughn, 1997), MATLAB 5.0 (Bramley, 1997; Conrad, 1997; Foster, 1997; Schaufelberger, 1997; Waynant and Fuller, 1997). Comparisons of some of these mathematical packages for science and engineering education have been discussed by Seiter (1992), Pattee (1995) and Bromilow (1997) but none of these comparisons considers all four packages.

Two studies on the teaching of first year undergraduate calculus courses, using Mathematica, report favourable outcomes (Park and Travers, Noss, 1995). Both courses were entirely computer-based, on-line with interactive text, giving students access to many examples. Students experienced calculus as a course in scientific measurement, calculation, and modelling, through the use of technology.

*Technology also makes it possible to present the subject as a highly visible, often experimental, scientific endeavour.*

(Uhl, 1995)

When using Mathematica for teaching chemical engineering concepts in process control, and reaction engineering, Dorgan and McKinnon (1996) found that students had mixed but generally positive reactions to its use. Articles featuring the use of symbolic algebra computing in Control Engineering were published recently to arouse a greater awareness of the potential offered to engineers by environments such as Mathematica and MATLAB (Munroe, Barker and Zhuang, 1997). Munroe emphasised that there are many areas where symbolic computing can offer significant improvements in the reliability and accuracy of results obtained.

This chapter is concerned with mathematical power tools, general purpose computer applications for mathematical calculations and symbolic algebraic manipulation. Their comparison and evaluation will be from the view point of an undergraduate

engineer seeking to solve mathematical problems, quickly and reliably and to communicate results. A direct comparison of each package with the others is given for each of the problems posed.

### 3.2 MATHEMATICAL TOOLS

**MAPLE™** performs computations that include symbolic algebra and numeric approximations, linear algebra, calculus, trigonometry, and differential calculations, infinite and indefinite integration, modelling, statistics, graphics, and produces program statements for a FORTRAN compiler. Maple is a symbolic manipulation language that clearly displays algebraic expressions especially useful for integration and differentiation (Heck, 1993).

**MATHEMATICA™** combines numerical calculations and symbolic manipulations into an interactive environment, coupled with graphic visualisation and a high-level programming language. The program is divided into a kernel, for the computation, and the front end, which provides the user-interface and input capabilities. Mathematica equations are stored and can be imported or exported in ASCII format, favouring high portability. The Mathematica interface enables users to organize text, graphics, computer output and pictures in a single 'notebook'. Included with Mathematica are standard functions and add-ons which allow the advanced user to perform complex mathematical analysis.

**MATLAB™** is an interactive, matrix-based system for scientific and engineering numerical computation and visualisation (Etter, 1993). The program operates with scalars, vectors, and matrices from expressions entered by the user. A variety of in-built functions can be used for displaying two or three dimensional colour graphics. The basic MATLAB package may be extended with different tool boxes designed for engineering specialities; System Identification, Optimisation, Control, Splines and Simulink. MATLAB is matrix based and particularly suited for signal processing (Orsak and Etter, 1995), digital communication, and control system design. A symbolic mathematics option in MATLAB uses the Maple kernel that extends its numerical capabilities to algebraic manipulation.

**EXCEL™** is a popular spreadsheet with limited symbolic capability but it is very effective for small engineering calculations. The wide range of in-built mathematical and statistical functions, the ease of interactive programming, ease of re-use and modification, rapid graph generation, and on-line help make it an efficient design and prototyping tool. Although Excel was designed for business purposes, it is a practical tool for scientists and engineers (Orvis, 1996).

### 3.3 ENGINEERING PROBLEM SOLVING

The challenge of solving real world problems provides the essence of engineering satisfaction. These mathematical applications facilitate the problem solving process to provide, quickly and efficiently, the required answers. To evaluate the usefulness



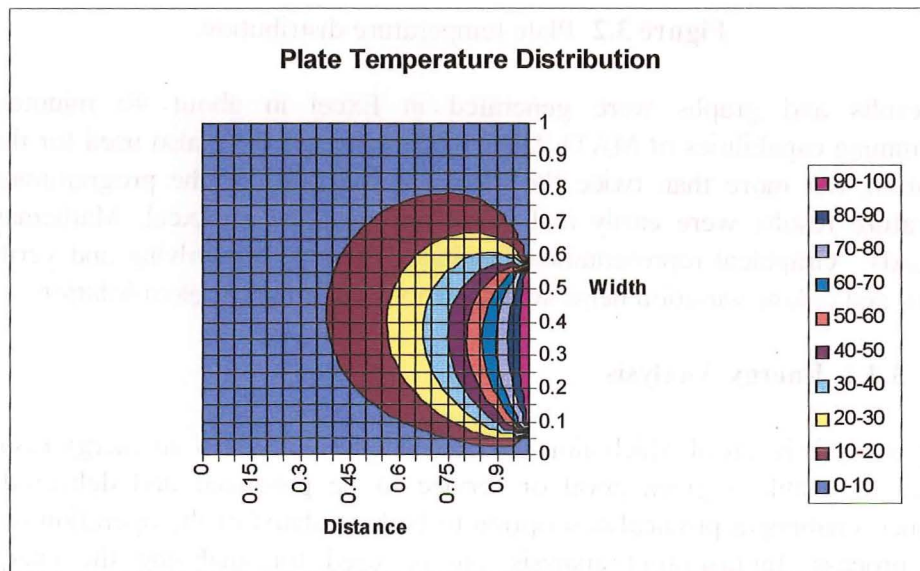
of these mathematical tools in the engineering problem solving process, six engineering problems were solved for comparison purposes using each of the four mathematical packages. The problems were the calculation and graphical display of a heat transfer calculation, the inversion of a large matrix as part of an input-output economic analysis, an illumination design needing a multi-variable optimisation, a Fourier series approximation, and two process control problems (one root finding for Bode design and the other requiring integration of a system of differential equations to evaluate the time response for the designed controller).

### 3.3.1 Two-Dimensional Heat Transfer

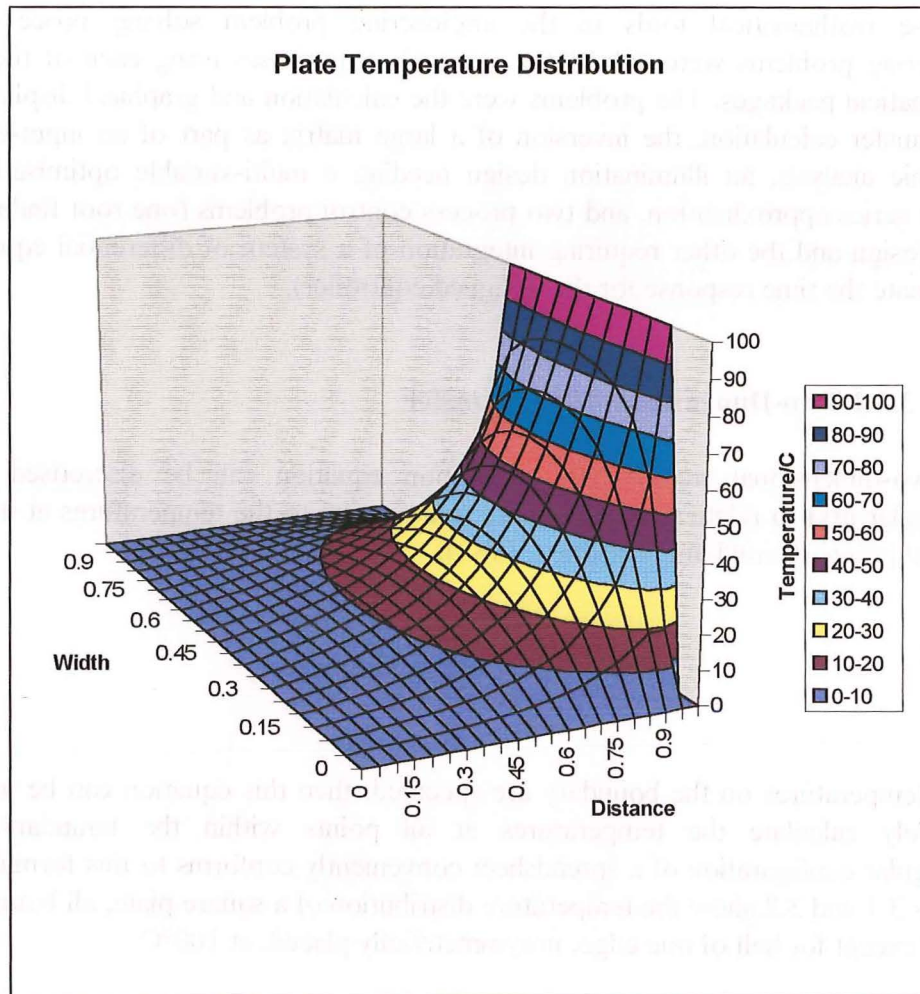
The two-dimensional steady-state conduction equation can be discretised on a rectangular grid to relate the temperature at any point to the temperatures at its four adjacent points (Gerald and Wheatley, 1984).

$$T_{i,j} = \frac{T_{i-1,j} + T_{i+1,j} + T_{i,j-1} + T_{i,j+1}}{4} \quad (1)$$

If the temperatures on the boundary are specified, then this equation can be used to iteratively calculate the temperatures at all points within the boundary. The rectangular configuration of a spreadsheet conveniently conforms to this formulation. Figures 3.1 and 3.2 show the temperature distribution of a square plate, all boundaries at 0°C except for half of one edge, unsymmetrically placed, at 100°C.



**Figure 3.1** Contour plot of plate temperature distribution from Excel



**Figure 3.2** Plate temperature distribution.

The results and graphs were generated in Excel in about 40 minutes. The programming capabilities of MATLAB and Mathematica were also used for the same calculation, but more than twice the time was required for the programming. The temperature results were easily and effectively graphed in Excel, Mathematica or MATLAB. Graphical representation facilitates the problem solving and verification process, and colour variation helps students to visualise the problem solution.

### 3.3.2 Energy Analysis

Energy analysis is a tool which aims to determine how much of an energy resource is required to enable a given good or service to be produced and delivered to its consumer, enabling a physical description to be formulated of the operation of a real-world process. Input-output analysis can be used for analysing the energy and environmental consequences of consumption. Peet (1991) gives an example using 80 sectors.

Mathematically, the problem was that data were collected in the matrix form

$$X=AX + Y$$

(2)

where X and Y were vectors of system inputs and outputs, and A was the technical coefficient matrix. Therefore

$$Y = X - AX = (I - A).X \quad (3)$$

where I is the identity matrix. Thus the system was analysed in the form

$$X = (I - A)^{-1} . Y \quad (4)$$

Computationally, an 80 by 80 matrix was required to be subtracted from the identity matrix and inverted. The matrix was easily inverted with MATLAB or Mathematica but Excel was unable to handle such a large matrix. However, it was important to have the data in the correct format before importing into both MATLAB and Mathematica.

Data were saved as plain ASCII text, and imported to Mathematica with

```
mymatrix=ReadList["c:\excel\energy.dat", Number, RecordList -> True]
```

and into MATLAB with

```
load c:\excel\energy.dat
```

Maple also has a “read” function for importing data but was unable to import the large data matrix .

After solving a large system of linear equations, an estimate of the condition of the computed solution is important to verify numerical accuracy. Maple and MATLAB have functions to estimate the condition number of the inverted matrix to provide this verification (Kahaner, Moler, and Nash, 1989). The condition number of the matrix above was 9.8 indicating that the inversion was relatively accurate, resulting in the loss of only one significance place.

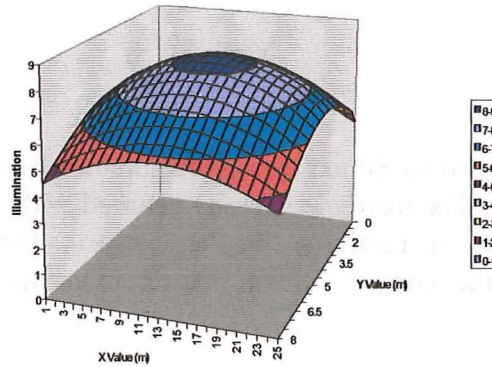
### 3.3.3 An Illumination Design by Optimisation.

An 8m by 12m rectangular concrete pad for a bicycle stand was to be illuminated with three lampposts, 200w and 7m high. Where should the three lampposts be positioned to maximise the illumination at the point of minimum illumination on the pad? If the power of the lamp j is  $P_j$ , its height  $h_j$ , and its co-ordinates are  $x_j$ ,  $y_j$ , then the total illumination from the N lampposts, at the point with co-ordinates x, y, is :

$$I = \sum_{j=1}^N \frac{P_j h_j}{\sqrt{[h_j^2 + (x - x_j)^2 + (y - y_j)^2]^3}} \quad (5)$$



In Excel, a grid was set up to calculate the illumination at any point on the pad, and plotted as a contour plot (Figure 3.3). Inspection showed that the minimum illumination point was at a corner, and the problem simplified by assuming that the optimal layout was with lampposts symmetrically down the longitudinal centre line, with one lamppost positioned at the centre. The Solver was then used to find the lamppost spacing to maximise the corner illumination.



**Figure 3.3** A 3D contour plot generated in Excel shows the illumination of the bicycle pad.

This problem could not be solved symbolically in Maple, Mathematica or MATLAB because of the complexity of the illumination equation (5).

### 3.3.4 Fourier Series Approximation

The dynamics of a process can be identified by forcing it with an input pulse, for example a flow of steam, and measuring the temperature response, typically also a pulse. As an example of the required mathematical processing, a typical input pulse

$$x = \exp\left(\frac{t}{5}\right) - \exp(t) \quad (6)$$

was approximated by the Fourier series:

$$f(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos \frac{2\pi nt}{T} + b_n \sin \frac{2\pi nt}{T} \quad (7)$$

where

$$a_0 = \frac{1}{T} \int_{-T/2}^{T/2} f(t) dt \quad (8)$$

$$a_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \cos \frac{2\pi nt}{T} dt \quad (9)$$

$$b_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \sin \frac{2\pi nt}{T} dt \quad (10)$$

The programming requirement was minimal to symbolically complete the integrations, and evaluate and plot the fitted function with a simple nested loop construction. MATLAB gave a solution in about 30 seconds using the following m.file:

```
t=0:0.01:30;
for p=1:4
T=50;
k=input('how many terms? ')
f1='exp(-t/5)-exp(-t)';
z=exp(-t/5)-exp(-t);
a0=eval(int(f1,0,T));
y=(1/T)*a0;
g='(exp(-t/5)-exp(-t))*cos(2*pi*n*t/T)';
h='(exp(-t/5)-exp(-t))*sin(2*pi*n*t/T)';
an=int(g,'t',0,T);
bn=int(h,'t',0,T);
for n=1:k
y=y+(2/T)*eval(an)*cos(2*pi*n*t/T)+(2/T)
    *eval(bn)*sin(2*pi*n*t/T);
end
subplot(2,2,p); plot(t,y,'w-',t,z,'w-.');
axis([0 30 -0.1 1])
title(['n= ',num2str(k)])
end
```

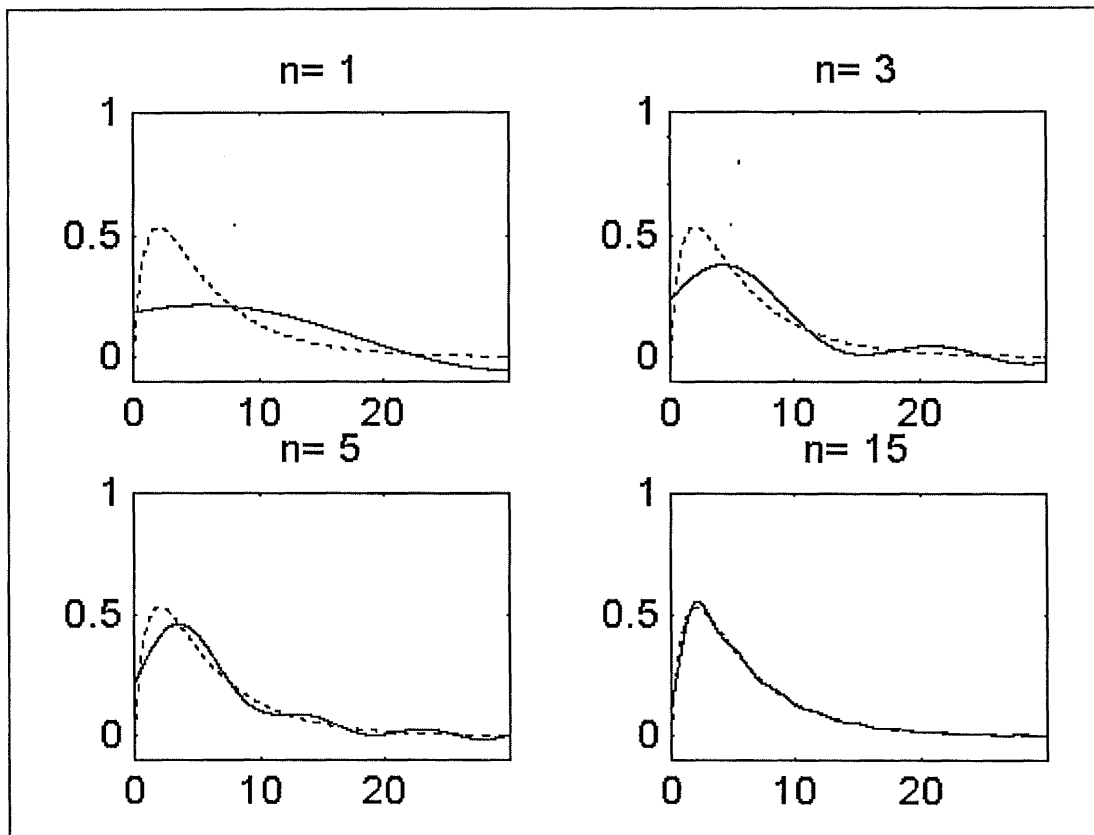


Figure 3.4. Graphical subplots of the input function using Fourier series

Graphical subplots, as in Figure 3.4, showed the quality of the approximation as the number of terms in the series,  $n$ , was increased. A similar problem was given to second year engineering students as an assignment in their mathematics course. MATLAB has proved very suitable for solving this problem as well as providing a different method for teaching mathematics. Students were given a model example and detailed explanation to minimise time spent learning MATLAB enabling them to concentrate on the mathematical problem.

Maple and Mathematica gave similar results to MATLAB, both taking slightly longer to run. Excel, without symbolic integration, was less convenient for this problem because the users needed to do the integration themselves.

### 3.3.5 Root-finding for Control System Design.

An ideal Proportional -Integral-Derivative controller, having Laplace transform,

$$\bar{m} = K_c \left( 1 + \frac{1}{T_i s} + T_d s \right) \bar{e} \quad (11)$$

where  $m$  and  $e$  are the valve position and error respectively;  $K_c$  is the proportional sensitivity; and  $T_i$  and  $T_d$  are the integral and derivative times, was to be designed for the fourth order process (Figure 3.5)

$$\bar{c} = \frac{1}{(T_a s + 1)(T_b s + 1)(T_c s + 1)(T_e s + 1)} \bar{m} \quad (12)$$

where  $T_a, T_b, T_c, T_e$  are time constants. The Bode stability criterion (Coughanowr and Koppel, 1965) requires the solution of the equation:

$$F(x) = \pi - \tan^{-1}(T_a x) - \tan^{-1}(T_b x) - \tan^{-1}(T_c x) - \tan^{-1}(T_e x) + \tan^{-1}\left(T_d x - \frac{1}{T_i x}\right) = 0 \quad (13)$$

for the angular frequency,  $x$ . This numerical approach, coupled with the calculation of the amplitude ratio, replaces the well-known graphical procedure (Coughanowr and Koppel, 1965) using Bode Diagram graph paper.

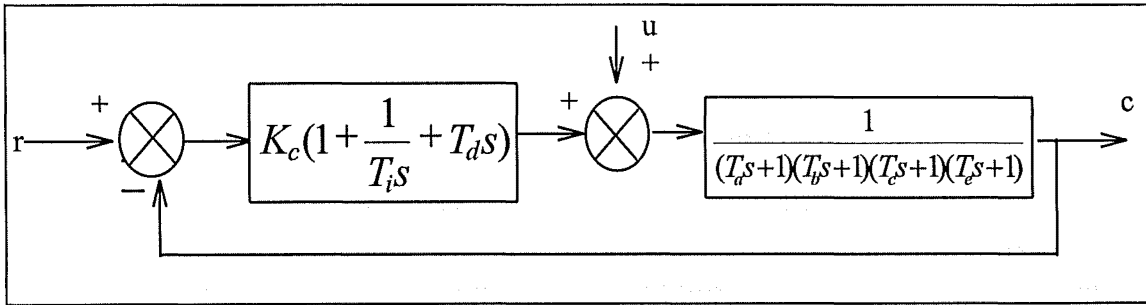
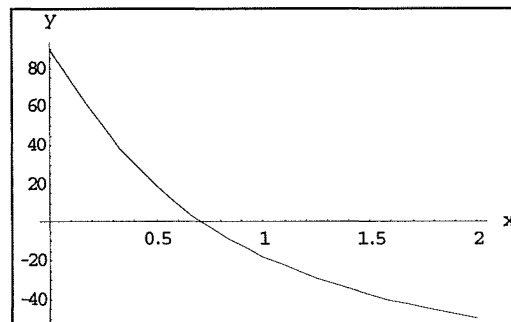


Figure 3.5 Control System Block Diagram

This root-finding problem was successfully and quickly solved by each of the three applications. The dialogue with Mathematica, with the Mathematica response indented, was:-

```
F := P - ArcTan[x Ta] - ArcTan[x Tb] - ArcTan[x Tc]
      - ArcTan[x Te] + ArcTan[x Td - RTi / x]
Ta = 1; Tb = 2; Tc = 3; Te = 4; Ti = 7;
RTi = If[Ti > 0.0, 1/Ti, 0.0];
Td = Ti / 4; P = π;
FindRoot[F == 0, {x, 0.5}]
Plot[F / Degree, {x, 0.002, 2},
      AxesLabel -> {"x", "y"}]
{x -> 0.716372}
```



Time (sec)

Figure 3.6. Mathematica output to confirm location of the angular frequency root.

Excel was used in two ways for this problem. Firstly, a Newton root finding method was derived by hand, taking about 50 minutes to calculate and verify the derivatives. Each successive iteration was a row of the spreadsheet. Alternatively, when the Goal-seeking command was used, only 5 minutes were required, similar to the time required for any of the other applications (Allen and Mackenzie, 1996).

However, in the full design procedure, the root-finding has to be repeated three times, with only minor modification. Each of the applications supports the construction of

user-defined functions, similar to functions, procedures or subroutines in procedural programming languages.

### 3.3.6 Differential Equation Solution for Time Response

The control system designed in 3.3.5 had the form shown in Figure 3.5. The engineering problem 3.3.6 was concerned with the response of this system to a change in set point,  $r$ , or to a disturbance,  $u$ , in order to confirm the control system design.

Differences between the nature of the mathematical applications became more evident when solving this problem. Maple and Mathematica provided the simplest solution. The relationships defining the closed loop transfer function were defined and solved automatically for the Laplace transform of the process variable,  $c$ . The inverse Laplace transform function (Abell and Braselton, 1994) available in both Maple and Mathematica gave the required time response within an elapsed time of 5 minutes.

The Mathematica dialogue and response were:-

```
<<Calculus`LaplaceTransform`
Ta=1; Tb=2; Tc=3; Te=4; Ti=7;
RTi=If[Ti>0.0,1/Ti,0.0]
Td=Ti/4; Nf=1; u=0; r=1; Kc=5;
G=Kc(1+RTi/s+Td s)
Df=(Ta s+1)(Tb s+1)(Tc s+1)(Te s+1)
c=FullSimplify[(Nf u+Nf G r)/s/(Df+G Nf)]
InverseLaplaceTransform[c,s,t]
Plot[%,{t,0,60}, Frame->True,
      FrameLabel->{"Time (sec)","Response"}]
```

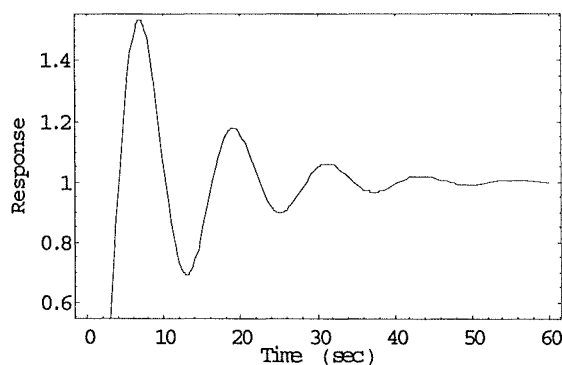


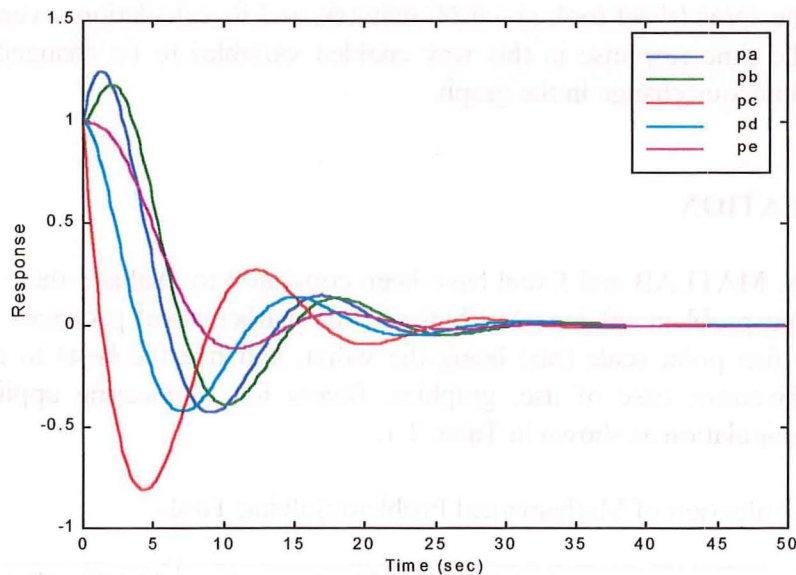
Figure 3.7 Time response in Mathematica - set point response.

A finite difference approach could have been taken in MATLAB, and one of its differential equation solvers, ODE23 or ODE45, used. However, since the problem was linear, the closed loop transfer function was rewritten in state space form, and the matrix exponential function used to calculate the time response.



The MATLAB m-file to set up the process matrix A, calculate its eigenvalues, and derive and plot the time response was:-

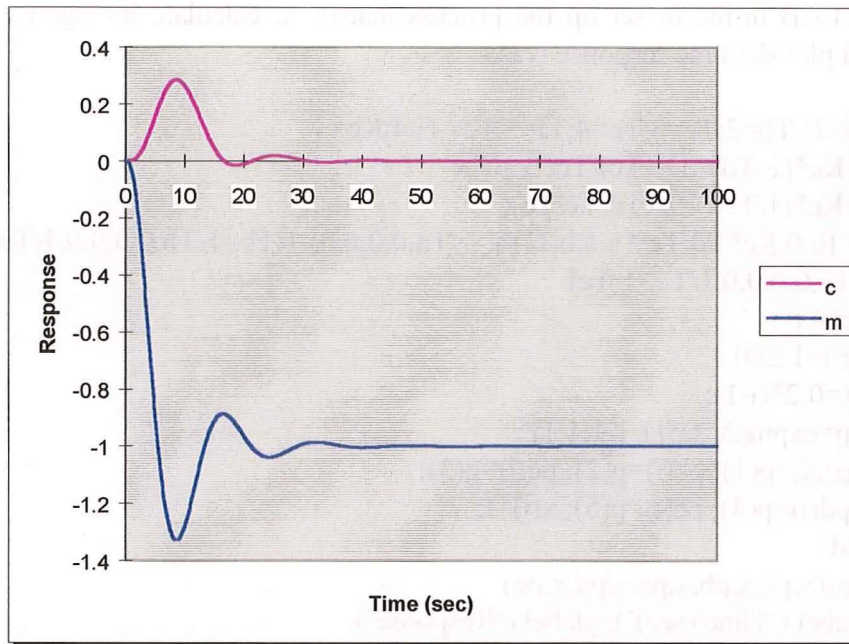
```
Ta=1; Tb=2; Tc=3; Te=4; Ti=7; Td=Ti/4; Kc=3
a=Kc*(1-Td*(Te+Tc)/Tc/Te)/Te;
b=Kc*(1/Ti-1/Te+Td/Te/Te);
A=[0,0,Kc*Td/Tc/Te,a,b;1/Ta,-1/Ta,0,0,0;0,-1/Tb,-1/Tb,0,0;0,0,1/Tc,-
1/Tc,0;0,0,0,1/Te,-1/Te]
eig(A)
for i=1:250
    t=0.2*(i-1);
    p=expm(A*t)*[1 1 1 1 1]';
    pa(i)=p(1); pb(i)=p(2); pc(i)=p(3);
    pd(i)=p(4); pe(i)=p(5); x(i)=t;
end
plot(x,pa,x,pb,x,pc,x,pd,x,pe)
xlabel('Time (sec)'), ylabel('Response')
legend('pa', 'pb', 'pc', 'pd', 'pe');
```



**Figure 3.8** Control system initial condition response.

The algebra required 100 minutes, setting up the MATLAB calculation a further 10 minutes, and the actual calculation and plotting took about 10 seconds.

Excel did not have differential equation support. A solution was obtained by deriving the differential equation for the controller and each of the first-order elements from its transfer function and using a finite difference approximation to provide a recursive relationship.



**Figure 3.9** Excel graph of time response to a unit step in the load,  $u$ .

Setting up the spreadsheet took about 60 minutes, and its calculation several seconds. Modelling the time response in this way enabled variables to be changed giving an almost simultaneous change in the graph.

### 3.4 EVALUATION

Mathematica, MATLAB and Excel have been compared to evaluate their application to engineering problem solving. Attributes of the mathematical packages were rated on a one to five point scale (one being the worst, and five the best) to assess their scope, intuitiveness, ease of use, graphics, fitness for engineering application and symbolic manipulation as shown in Table 3.1.

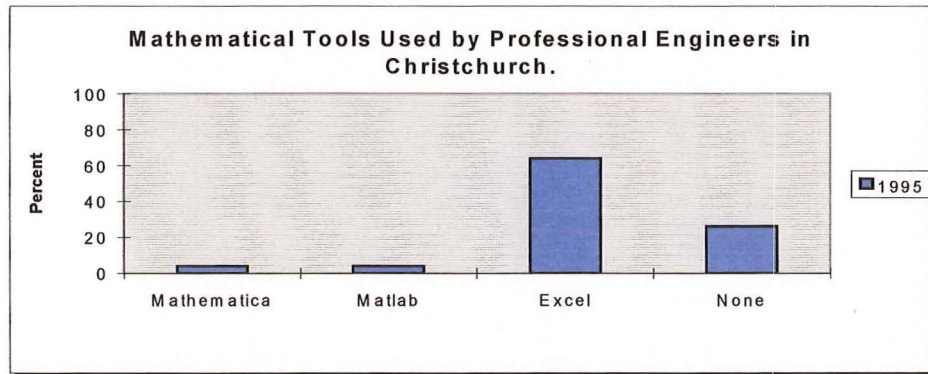
**Table 3.1** Evaluation of Mathematical Problem Solving Tools

	Maple	Mathematica	MATLAB	Excel
Scope	4	5	5	3
Intuitiveness	4	5	5	4
Ease of use	3	5	4	3
Graphics	3	4	4	5
Fitness for engineering application	4	5	5	3
Symbolic manipulation	5	5	4	0
<b>Total</b>	<b>23</b>	<b>29</b>	<b>27</b>	<b>18</b>

All four mathematical packages are powerful problem solving tools. In this evaluation, Mathematica was ranked ahead of MATLAB with Maple following a close third and Excel fourth. However, such comparisons are subjective, and the differences between the packages were small.

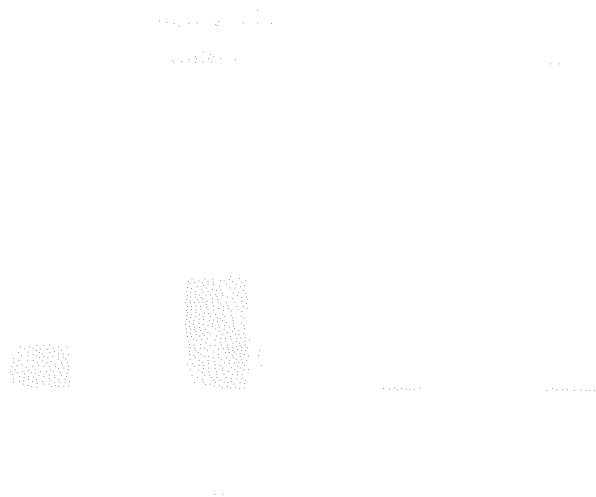
### 3.5 ENGINEERS AND MATHEMATICS

Do professional engineers use computer applications to solve the mathematical problems they encounter every day in New Zealand? A telephone survey of fifty professional engineers was conducted in the local Christchurch area to investigate what mathematical packages were being used in professional practice in 1995. Sixty-four percent used Excel in their daily work, 4% used Mathematica, 4% used MATLAB, and 28% did not use any of the mathematical applications (Figure 3.10). Users of Mathematica and MATLAB all had an affiliation with a university.



**Figure 3.10** Mathematical tools used by Christchurch Engineers.

The best tool depends on individual needs and time spent learning the applications will reap the benefits of these powerful mathematical tools. How to incorporate these powerful mathematical tools in our graduate and undergraduate courses is a key issue for engineering educators.



#### **4.1 PROBLEM SOLVING**

Problem solving is an activity which engineers are engaged in every day, but not many professional engineers are taught problem solving strategies as part of their undergraduate education.

In the literature, writers on problem solving such as Wickelgren (1938), and Polya (1945, 1962) were particularly concerned with problems based on mathematics and logic, whereas de Bono (1967, 1968, 1969, 1973), and Wertheimer (1961) were concerned with more open-ended problems and lateral thinking.

Some research on problem solving was based upon computer-aided modelling of mental processes known as artificial intelligence (AI) (Johnson-Laird and Watson, 1977; Newell and Simon, 1972; Kahneman, 1986). Other work was based upon problem solving experiments with human subjects (Johnson-Laird, 1983), and studies of thinking (Wittrock, 1986). Much of the research has been concerned with logical confusion and errors. As yet, not a lot of research has been done on the teaching of problem solving skills in higher education.

The teaching of problem solving skills to undergraduate engineering students has been described by Woods (1985), Fogler (1995), Ko and Hayes (1994), and Allen et al (1996). From an industrial point of view Maul and Gillard (1996), emphasise that problem solving is a crucial skill for engineers when they work in manufacturing. They believe that the time to teach this skill is critical in the professional development of an engineer, and should be immediately following graduation when the engineer is confronted with what seems to be an unsolvable problem.

#### **4.2 A NEW APPROACH TO ENGINEERING DESIGN**

In traditional design teaching, the design process is often broken up into a number of incremental steps: defining the task, goal setting, establishing a concept, defining the constraints, setting the specifications, listing the alternatives, evaluating and selecting the best alternatives, formulating an appropriate mathematical model, calculating, modifying, costing, drawing, constructing, testing and finally commissioning. A general structure of the design process, once recognised and defined, is then adapted as a design strategy for future projects. While the approach of retrospectively studying successful design projects, recognising the various areas of activities, their logical sequence and applying it to new projects works well, it relies heavily on experience for a successful outcome. However, it is exactly this experience in application that our undergraduate students lack. A practical design strategy is more appropriate for novice engineers.



Design has been taught to our Chemical Engineering undergraduates for many years by a traditional case studies approach which involved dissecting the design process into its various elements, imparting relevant knowledge by formal lectures, and demonstrating how experienced engineers have designed successful systems. It was hoped that this approach would imbue the students with sufficient knowledge and skills to become confident designers.

These efforts to teach design led to the realisation that competence in design seemed to be “caught” by a handful of students who rose to the challenge and were able to apply skills, knowledge and other personal attributes, often with outstanding results. The recipe for success seemed to combine such ingredients as organisation, lateral thinking, computation, practical experience in workshop skills and an ability to think in abstract terms. The “special talents”, which every student possesses, need to be developed and honed to a sharper edge.

In recent years, a problem solving approach, similar to that of (Woods, 1983) has been adopted for the teaching of third year engineering design. A problem solving foundation to engineering design provides students with the necessary skills and confidence to be able to tackle any problem, design or otherwise, without feeling hindered by lack of direct experience in the particular topic.

#### 4.3 TEACHING AIMS AND OBJECTIVES

Teaching aims and objectives were:

- To provide an innovative, interactive teaching and learning experience for third year Chemical Engineering students.
- To develop students understanding of different problem solving strategies.
- To introduce students to *Strategies for Creative Problem Solving Modules*.
- To monitor students learning of problem solving concepts using on-line assessment.
- To evaluate on-line assessment.

#### 4.4 RESEARCH AIMS AND OBJECTIVES

Research aims and objectives were to evaluate:

- Students' reaction to *Strategies for Creative Problem Solving Modules* as part of their third year chemical engineering design course.
- Students responses to working in pairs for problem solving.

## 4.5 CREATIVE PROBLEM SOLVING MODULES

Students learn best when they directly experiment with subject matter and are actively involved with the material (Felder, 1988). Interactive computer instruction provides active learning and allows students to

*...review and demonstrate mastery of the material at his/her own pace, [and] provides them with immediate feedback to their responses.*

Fogler, (1992)

Interactive computer-based learning depends on software that is easy to use, maintains a focus on the concepts, has minimal tediousness, promotes learning and gives individual guidance. *Strategies for Creative Problem Solving*, a collection of interactive computer modules developed by Fogler (1995), were used to supplement problem solving lectures. Additional features in some modules, such as the use of graphical animation and entertaining motivators, were included to increase student interest in, and motivation for, the module content (Snow, 1987).

### 4.5.1 Introduction to Problem Solving.

This module provides the user with the motivation to use creative problem solving strategies. Topics in this module include the characteristics of effective problem solvers, fear of failure, the need for risk taking, paradigm shifts, having a vision, a problem solving heuristic, creative thinking and working in teams. The introduction section presents these topics as well as their application to a contamination problem in a municipal water supply system.

### 4.5.2 Problem Definition Techniques.

The goal of this module is to help the user properly define the problem. Several techniques are used to better define the problem statement; for example the Dunker Diagram, the McMaster Five Point Strategy (Woods, 1983), the Present-State Desired-State technique (Higgins, 1967) and the Statement-Restatement technique. The user reviews the methods of problem definition in two examples; problems at a flashlight manufacturing plant are analysed with the McMaster Five Point Strategy and a second example involves a grocery store freezer door fogging up and blocking customers' view of contents.

### 4.5.3 Brainstorming - Methods of Solution Generation

This module helps the user to generate original yet applicable solutions to a specific problem through the brainstorming process. The review section introduces the basic techniques and ideas for improvement including Osborn's checklist (Adams, 1974); random word stimulation, futuring, conceptual blockbusting and using other people's views. These methods are illustrated through specific examples. To test these techniques, the user is asked to brainstorm a list of synonyms for the word "money". Once the user is finished, the user's list is compared to one generated by a group of college students. In the interactive session, the user selects at least two brainstorming

sessions chosen from a list of five possible scenarios. The topics for these scenarios range from encouraging recycling in a community to preventing zebra mussel infestations on power plant water intake pipes. For each scenario, a detailed problem statement is given, as well as a few example solutions to get the user started.

#### **4.5.4 Situation Analysis**

Kepner-Tregoe (Kepner, 1981) situation analysis is used to assess a situation in which multiple problems occur simultaneously. In the introduction, a situation analysis table is presented and the concepts of timing, trend, impact, and choosing the appropriate problem analysis technique are introduced. The main scenario is based on an explosion of a gas tanker truck in a mid-western town in USA. The user is part of a consultation team that is called in to analyse and develop an order of priority, and to prepare a situation analysis chart for the explosion scenario. Opportunities for investigation are provided via computer simulation for interaction with witnesses, the police, the fire chief and co-workers. Before leaving the module, users can compare answers with a model situation analysis chart.

#### **4.5.5 Problem Analysis - Finding the True Cause**

The concept of problem analysis is introduced in this module, as well as the four dimensions of a problem; identification, location, timing and magnitude. A malfunctioning piece of laboratory equipment is used as an example for problem analysis. The Kepner-Tregoe problem analysis technique is presented in a chart format to illustrate problem identification. The user is then given an assignment as a paint engineer to determine the cause of paint defects on cars. The investigation is aided by the opportunity to speak with different workers involved with the process, and a Kepner-Tregoe chart has to be completed.

#### **4.5.6 Decision Analysis - Choosing the Best Solution**

Kepner-Tregoe decision analysis chooses the best solution from a number of alternatives. A sample scenario is used to explain the decision analysis technique. The decision analysis chart is broken into three parts; musts, wants, and adverse consequences. For the main assignment, the user is asked to interview job applicants for a position as a process engineer, and to complete a decision analysis chart to select the best applicant.

#### **4.5.7 Potential Problem Analysis - Avoiding Future Problems**

Potential problems should be anticipated and analysed before they happen. Three parts of the potential problem analysis; possible causes, preventative action, and contingent actions; are explained in the introduction. The user then has a choice of scenarios, either a cross-country road trip or preparation for an interview, which are used to review the techniques. The main scenario is based on the 1993 World Solar Car race, Sunrayce'93 (Morrison, 1993). The background of the race is presented with additional explanation of relevant technology, including the solar cell mechanism and the importance of gear ratios in power train design. A potential



problem analysis chart for the event is prepared by the user to determine problems that might occur during a race and their prevention.

#### **4.5.8 Planning - Implementation of Solutions**

Gantt Charts (Gantt, 1910), Critical Path Analysis (Lockyer, 1991), Deployment Charts and Budget Proposals are introduced as tools to aid planning. These four techniques are illustrated in two introductory scenarios: planning the ergonomic design of an office and planning a student conference. In the interactive section of the module, the user is part of a team participating in a student competition to build a one-tenth scale model of a steel bridge. Each of the planning techniques is then applied to generate a Gantt chart, a critical path chart, a deployment chart, and a budget for the project.

#### **4.5.9 Evaluation - Solution Evaluation Techniques**

The importance of continually re-evaluating a solution throughout the course of a project is emphasised. The technique presented is the Evaluation Checklist, illustrated by the near disaster of marketing New Coke. The example demonstrates the use of the Evaluation Checklist to prevent millions of dollars from being wasted. In the interactive scenario, the user is presented with the problem of a paper mill company which plans to expand its production capacity. The user is given the opportunity to talk to other employees in the company and to gather the necessary information to evaluate the proposed expansion. Findings are submitted to the Project Supervisor for immediate feedback.

### **4.6 COURSE ORGANISATION, ASSESSMENT AND EVALUATION**

The problem solving section of the design course consisted of 9 one-hour sessions with about the same time devoted to working through set problems. This allocation of time was just sufficient to introduce the 40 third-year Chemical Engineering students to the basic concepts, to give them the experience of applying these new skills and to expand their confidence in analysing and solving new problems independently.

The importance of communication and working in teams in the process of problem solving was emphasised. The Whimbey and Lochhead (Whimbey, 1980) technique of attacking problems was taught, with one of a pair playing the role of problem solver and the other the listener, and alternating roles. The first problem solving assignment, worth 10%, gave randomly selected student pairs the opportunity to apply this technique to a set of problems taken from the McMaster problem solving program (Woods, 1983).

The second assignment was based on the Fogler interactive computer modules and was also worth 10% of the course marks. Each pair of students was assigned two of the interactive computer modules to complete each week for three weeks - six modules in total. At the completion of each module, a computer-generated performance score was recorded by students and handed in as part of their

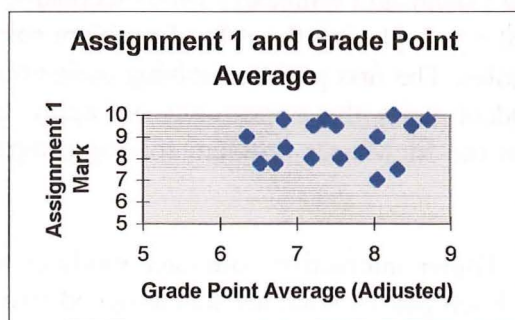
assessment. Students were given the option of repeating the modules as often as they wished to improve their score, and some did, with their best score being credited. A questionnaire completed by students at the end of the design course provided an evaluation of Whimbey pairs, and the problem solving modules.

#### 4.7 RESULTS AND DISCUSSION

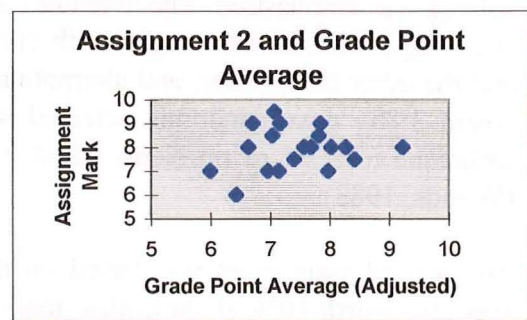
The distribution of marks in the first assignment was high with a skew towards a possible score of ten. When working through the McMaster problem set, a number of students showed interest in the Whimbey pair concept, with this interest being reflected in obtaining full marks. Table 4.1 is a sample of students' comments taken from the questionnaire.

**Table 4.1** Students' comments after experiencing Whimbey's method of problem solving.

Problem Solver	Listener
"As the problem solver, I found that when solving problems, I tend to like to put things into mathematical equations"	"In problem solving you must read the question carefully, jot down any conditions, and then determine what the problem is asking you to solve"
"For the basic problem, I did write out more than I usually would. It was fun and I would like to do more of it"	"The hardest thing was not to get carried away and tell the problem solver the answer when I knew it"
"I think I work too much out in my head and tend to rush to give an answer"	"It was a much easier task to be the listener than the solver".
"I tend to attack problems head on, noting down all the information supplied as I read it through"	"It's good to try to show the other person a different way of thinking"
"I enjoyed solving these problem".	"Being the listener is not an easy task!"
"The help of a listener was very useful; their ideas and reasoning are often very different and it's good to compare and see their point of view.".	"Listening is generally not too hard - often the solver doesn't vocalise everything".



**Figure 4.1** Marks for Assignment 1 compared with adjusted GPA.

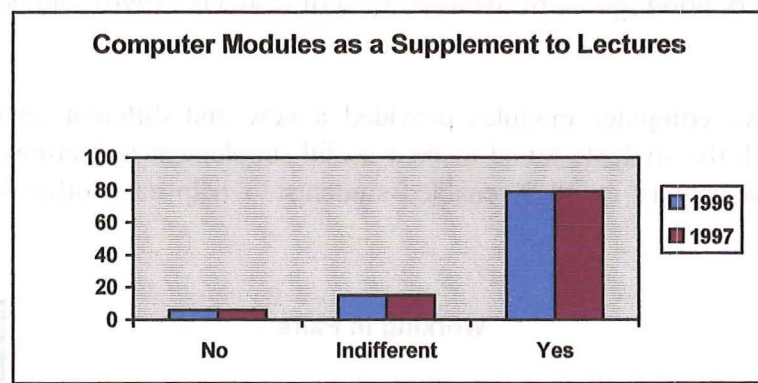


**Figure 4.2** Marks for Assignment 2 compared with adjusted GPA.

The assessment mark for each group for two assignments has been compared (Figures 4.1 and 4.2) with the group adjusted Grade Point Average for the 1996 end of year examinations. The Grade Point Average (GPA) for each student is the average grade for all papers sat by that student in the 1996 annual examinations, and is their final overall academic achievement in that year. The GPA is assigned a numerical equivalent ranging from 9 for an A+ to 2 for a C. The adjusted Grade Point Average (GPAJ) was recalculated to align means and standard deviation with marks out of 10.

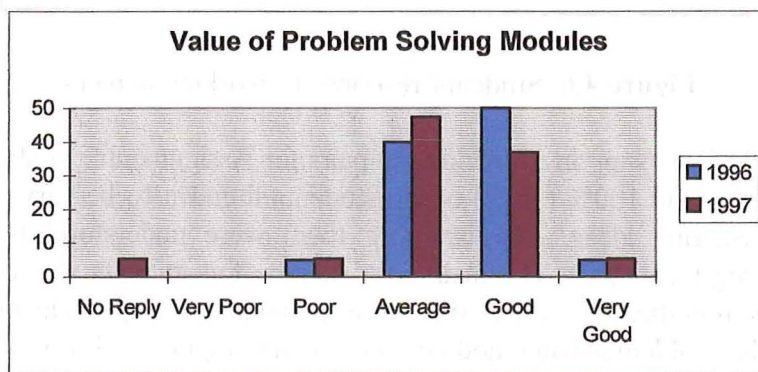
Statistical analysis showed no significant relationship between problem solving performance and previous academic results for the two problem solving assignments. A reason for the result is that problem solving is a different skill from conventional academic performance. In addition, the results can be explained by differences in testing procedures. The problem solving assignments were power assessments (without direct time constraints) whereas examinations were speed tests (with stringent time restriction to the examination time).

Students' responses to the questionnaire on the usefulness of the computer modules, their rating of the interactive problem solving modules, their opinion of computer-based assessment and working in pairs are shown in Figures 4.3 to 4.6.



**Figure 4.3** Students' response to the computer modules as a supplement to lectures.

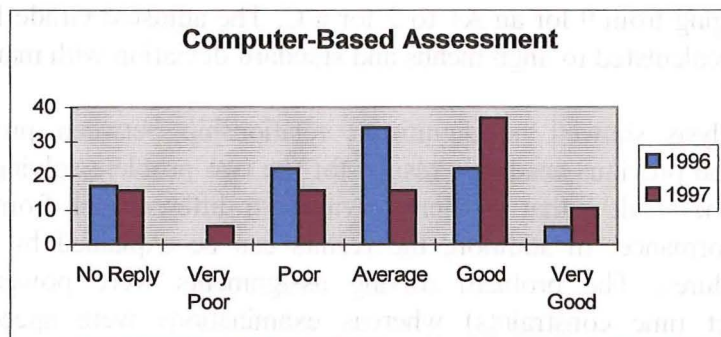
Figure 4.3 shows that 78% of the students in 1996 and 1997 found the computer modules to be a useful supplement to lectures, while 10% were indifferent and 2% did not find them useful.



**Figure 4.4** Students' opinion of the value of the interactive problem solving modules.



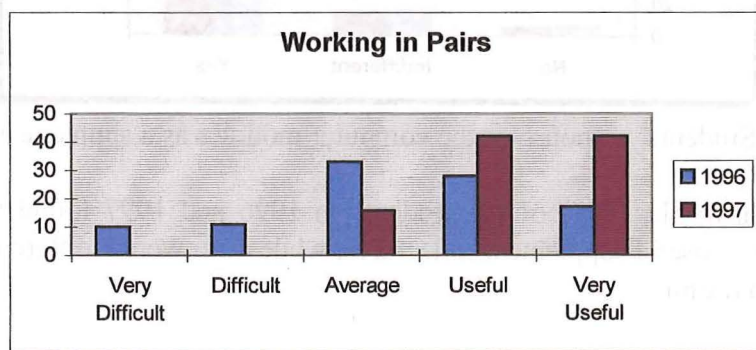
In Figure 4.4, students' rating of the interactive problem solving modules indicated a positive response with less than 5% (1996) and less than 10% (1997) rating them as worse than average. Students responded well to the practical problems in the computer modules, helping to understand different problem solving techniques.



**Figure 4.5** Students' response to computer - based assessment.

Computer-based assessment was introduced to students in the course last year. Figure 4.5 shows that 61% (1996) and 63% (1997) of these students found this form of assessment very good, good, or average. 17% of students (1996) did not respond to this question.

The interactive computer modules provided a new and different environment for learning which the students found to be a useful supplement to lectures. Working in pairs for Assignments 1 and 2, enabled students to help each other with problem solving.



**Figure 4.6** Students' response to working in pairs.

Most of the students supported working in pairs for Assignment 2 as shown in Figure 4.6, 78% (1996) and 100% (1997) of the students found it to be very useful, useful, or average. Sharing ideas and discussing them were understood to be valuable problem solving techniques. The main disadvantage for students was having to plan time to work together. In 1997, there was a significant improvement in students' opinions of the problem solving modules, as shown in Figures 4.3-4.6.

## 4.8 FURTHER RESEARCH QUESTIONS

As well as the student questionnaire, four further questions were examined:

- How did students perform on each computer module?
- Is there a gender difference in problem solving skills of our Chemical Engineering students?
- Is there a relationship with problem solving skills and other subjects?
- Does the academic ability of Chemical Engineering students vary from year to year?

### 4.8.1 Assessment On Each Module.

All students' marks for each module were assessed on-line. Table 4.2 and 4.3 show the mean and standard deviation for each module. In the 1996 group, all students scored full marks on the Introduction module, whereas this module ranked 6<sup>th</sup> for the 1997 group. Planning, Problem Analysis and Brainstorming ranked the lowest three scores for both 1996 and 1997 groups. The reason being that these students lack experience in such matters.

**Table 4.2** Students' assessment on each computer module in 1996.

	Mean Score (Percent)	Standard Deviation (Percent)
Introduction	100	1
Potential Problems	98	6
Situation Analysis	93	7
Evaluation	89	17
Define	86	15
Decision Analysis	73	16
Planning	68	21
Problem Analysis	65	22
Brainstorming	41	25

**Table 4.3** Students' assessment on each computer module in 1997.

	Mean Score (Percent)	Standard Deviation (Percent)
Situation Analysis	94	7
Define	92	10
Evaluation	87	21
Potential Problems	86	9
Decision Analysis	85	10
Introduction	83	11
Problem Analysis	76	20
Brainstorming	65	27
Planning	64	7

### 4.8.2 Gender Difference

**Table 4.4** Comparison of male and female third year Chemical Engineering students' marks for problem solving assignments.

	1996		1997	
	Male	Female	Male	Female
Number of Students	30	11	25	11
Mean	8.28	8.30	7.22	7.36
Standard Deviation	0.68	0.81	0.61	0.48

Female students mean marks for problem solving were slightly better than their male colleagues (Table 4.4). In 1996, the difference was only 0.02 and in 1997, the difference was 0.14; not a very significant difference.

### 4.8.3 Relationship With Other Subjects

Marks for 1997 third year Chemical Engineering students' problem solving assignments were correlated with final marks of second year subjects for 1996. The Course codes are shown in Table 4.5 and correlation results ( $r^2$ ) in Table 4.6.

**Table 4.5** Course Codes for second year Chemical Engineering students.

ENCH202	Engineering Materials 1 (Chemical)	ENCH252	Thermodynamics
ENCH211	Engineering Design 1 (Chemical)	ENCH253	Transfer Operations 1
ENCH250	Chemical Process Technology	ENGR250	Engineering Computation
ENCH251	Process Engineering	MATH	Mathematics

**Table 4.6** Correlation of subject and problem solving marks.

	ENCH 202	ENCH 211	ENCH 250	ENCH 251	ENCH 252	ENCH 253	ENGR 250	Math
$r^2$	0.173	0.00	0.08	0.053	0.08	0.049	0.028	0.133

Overall, there was very little correlation with marks attained for problem solving and second year subjects. Engineering Materials 1 (Chemical) and Mathematics gave the "best" correlation scores with an  $r^2$  of 0.173 and 0.133 respectively. But even these coefficients are so low that there was no correlation between problem solving assignments and regular courses for this student population.

#### 4.8.4 Academic Ability - Bursary Marks

Prior to entering University, students sit the entrance Bursary examination. Bursary marks for all 1996 and 1997 third year Chemical Engineering students were obtained from the University records.

Comparing the Mean and Standard Deviation of the two groups (Table 4.7), the 1996 group had a mean mark of 359 and the 1997 group, a mean mark of 343; both standard deviations were similar - 46 and 48 respectively.

**Table 4.7** Bursary Marks for 1996 and 1997 third year Chemical Engineering students.

	1996	1997
Number of Students	33	46
Mean	359	343
Standard Deviation	46	48
Minimum	270	235
Maximum	460	423

In Table 4.7, the 1996 group's mean mark was higher than the 1997 group by 16 marks. However, the marks obtained in the problem solving assignments by the 1997 group were higher than the 1996 group supporting the premise that problem solving skills are independent of academic skills.

Strategies for creative problem solving are techniques that can be learned. It is imperative that our graduates have the necessary skills and strategies to deal confidently with new situations and problems encountered in their professional careers.





### **5.1 THE AMOCO PROJECT**

The Amoco Project is a computer-based simulation exercise used as the final assignment for the third year chemical engineering design class. This project provided an industrial based example for planning, decision making and problem solving. In recent years the teaching of design in this course has adopted a problem solving approach integrated into formal lectures (Allen, et al, 1996; Woods, 1985).

Computer-based instruction has the potential to stimulate teaching and learning. Engineering courses can be enhanced by computer simulation and computer-assisted instruction. Seader (1989) comments:

*" The microcomputer is the most powerful learning device since the printed textbook. If the computer is used properly, individual interactive learning is economically possible for the first time".*

Computer simulation offers significant advantages (Squires et al, 1991):

- Processes that are too large, complex or hazardous for the university laboratory can be simulated on the computer.
- Realistic time and budget constraints can be built into the simulation, giving students a taste of "real world" engineering problems
- Emphasis of the laboratory exercise can change to experimental design and data analysis rather than the practical collection of data.
- Computer simulation is relatively inexpensive compared with the cost of building and maintaining expensive experimental equipment.

Problem solving techniques were taught in the first part of the design course. The Amoco project followed providing a practical design problem in which students could apply these problem solving techniques.

### **5.2 TEACHING AIMS AND OBJECTIVES**

Teaching aims and objectives were:

- To introduce students to the Amoco computer simulation model of a large chemical engineering process.
- To give students practical experience of important budgetary and time constraints when designing and operating a chemical engineering plant.

- To provide students with a real world example for planning and decision making as part of their design engineering course.
- To promote an active learning environment for application of chemical reaction theory to the Amoco pilot plant through the computer simulation model.
- To encourage co-operative learning by grouping the students.
- To give students the opportunity to devise a control scheme that will bring the reactor to steady state by adjusting variables in the computer simulation model.

### 5.3 RESEARCH AIMS AND OBJECTIVES

Research aims and objectives were to evaluate:

- Students reaction to the Amoco computer simulation model as part of their third year chemical engineering design course.
- Students response to working in groups

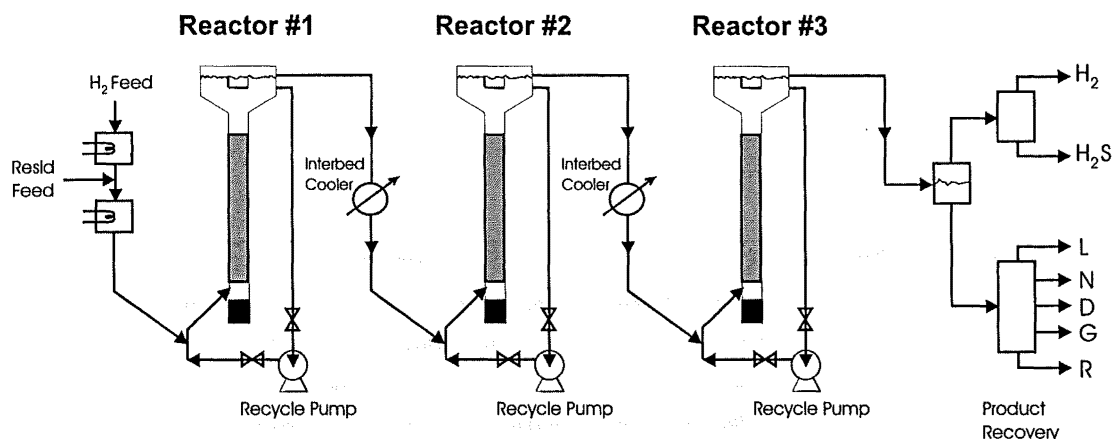
Students have many different learning styles and for effective learning “...*how subjects perceive their learning environments has long been accepted as having a significant influence on the quality of students’ learning outcomes*” (Jones, 1983).

### 5.4 THE AMOCO PROJECT

Amoco Resid Hydrotreater simulation (Figure 5.1) is one of several computer simulation models developed at Purdue University (Squires et al, 1991). The scenario is that Amoco have acquired delivery of a crude oil feedstock which is high in sulphur. Treatment of this crude oil will produce residual oil with 5 weight percent sulphur which needs to be upgraded to more valuable products in Amoco’s Texas City plant.

A mixture of heavy, high sulphur hydrocarbons (resid oil) is fed into the pilot plant which upgrades it by:

- breaking the long carbon chains to form smaller chains
- adding hydrogen to increase the saturation
- removing sulphur in the form of  $\text{H}_2\text{S}$  gas



**Figure 5.1** Amoco hydrodesulphurization pilot plant.

Each of the liquid components Resid (R), Gas Oil (G), Distillate (D), Naphtha (N), and Light Gas (L) is a complicated mixture of many different chemical species, which are treated as “pure” components to predict the performance of the reactors in the computer model. R, G, D, N, and L are characterised by their average boiling points and their sulphur content.

## 5.5 THE COMPUTER SIMULATION MODEL

The Amoco computer simulation model (ASM) was designed to run on a Sun 3/60 workstation. At the University of Canterbury the Amoco program was set up on our main UNIX system, and linked to the School of Engineering computer network running X Windows on the opposite side of the Campus. This was the first time the Amoco Simulation has been run with a Sun server connected to PC based display systems across a network. In 1997, the simulation has been running with less problems under Windows NT as the PC operating environment, compared to 1996 under Windows 3.1. However, Phase 2, developing a control strategy for the industrial plant, needed Sun workstations for the graphical display and output of six variables.

### 5.5.1 USE OF THE AMOCO SIMULATION

The ASM was the design project for our third year Chemical Engineering students. Eight two hour laboratory periods are allocated for the project based on the ASM. In the first laboratory session, students were given a written description of the process, and viewed a 20 minute video produced by Amoco to give some background to the plant. Details of the problem were given to the students so they could perform experiments using a simulation of a laboratory reactor before testing their results on a simulation of a pilot plant. The project was defined in two phases (see Table 5.1):

Phase 1: Initially, simulated laboratory and pilot plant reactor facilities were used to run a series of steady-state experiments to gather data on the behaviour of the resid feed stock. This data was used to predict the performance of an industrial scale plant using the model of the pilot plant.

Phase 2: Secondly, a start up procedure was developed for the industrial plant by running a dynamic simulation of the pilot plant.

**Table 5.1** Amoco Project Timetable

Week	Phase	Activity
<b>1</b>	<b>1</b>	Introduction, background to project, Video plant tour
<b>2</b>		Process Definition, Introduction to computer model, hands on practical session (No charge)
<b>3</b>		Planning the strategy and design of experiments. Practical session running the computer model (Charging started from this session).
<b>4</b>		Practical session, gathering data from computer model.
<b>5</b>		Practical session, analysis of data from computer model
<b>6</b>		Practical session, write interim report
		<b>Interim report due at end of Week 6</b>
<b>7</b>	<b>2</b>	Control strategy, problem definition, practical session running the simulation
<b>8</b>		Problem solution, run simulation, write final report
<b>9</b>		<b>Final Report due at end of Week 9</b>

### 5.5.2 BUDGETARY CONSTRAINTS

Contributing to the sense of the “real world” nature of this project, students are required to work within a budget. For example, students are given a *simulated* budget of US\$150,000. Table 5.2 lists the time and money required for various tasks related to the Amoco project. Note that the students were charged a fee each time they sought help from the “consultant” (i.e. the lecturers). Initially students were very concerned about the budgetary constraints; they had not previously encountered such limitations for their projects.

These constraints were to ensure that students think about the problem and carefully plan their laboratory experiments to:

- gather data about the chemical reactions (e.g. rate constants).
- gather data to calculate the rate of catalyst deactivation.

**Table 5.2** Expenses for the Amoco Project

Initial preparation and start-up of pilot plant (includes cost of initial change of catalyst)	5 days	\$75,000
Replacement of catalyst in pilot plant	3 days	\$50,000
One pilot plant run (includes labour, materials, analysis, etc.)		
Three reactors in series	24 hours	\$4,500 <sup>1</sup>
Two reactors in series	24 hours	\$4,000 <sup>1</sup>
One reactor	24 hours	\$3,500 <sup>1</sup>
One laboratory reactor run (includes catalyst replacement)	24 hours	\$500 <sup>1</sup>
Consultation	Per Session	\$500
<sup>1</sup> Multiply by 1.5 for Saturday runs; by 2.0 for Sunday runs		

### 5.5.3 PLANNING STRATEGY

In the third session students, in their groups, planned a strategy for the design of their experiments. A planning strategy was necessary as students were faced with a number of choices regarding the type and numbers of reactors, the catalyst age, the feed composition, the reactor temperature. In making these choices, they had to keep in mind a number of constraints:

- It takes five days and \$75,000 (half the budget) to clean and prepare the pilot plant.
- It costs another three days and \$50,000 to replace the catalyst in the pilot plant; obviously, the students cannot afford to change their minds on the catalyst selection.
- Each pilot plant run takes twenty-four hours<sup>2,\*</sup>.
- The pre-exponential factors  $a_1, \dots, a_7$  need to be measured in the pilot plant.
- Other constraints can be obtained from the laboratory reactor, but their values must be checked in the pilot plant.

It made sense to use the laboratory reactors as much as possible, since they were considerably cheaper to operate than the pilot plant. In addition, the laboratory reactors used so little catalyst they were easily refilled, permitting students to take data related to catalyst age. However, all constants that were determined from the laboratory data needed to be finally checked in the pilot plant.

---

<sup>2</sup>Simulated Time

### 5.5.4 THE INTERIM REPORT

Students were instructed to gather data and analysis for the interim report as follows:

- Determine the values of the pre-exponential factors for the hydrogenations
- Check the activation energy for one of the hydrogenations
- Determine the pre-exponential factor  $a_8$  and  $E_8$  for the desulphurization
- Check the form of the catalyst deactivation equation and measure the catalyst deactivation rate.

Students used Excel™ to fit a least squares regression line for the Arrhenius plot to find the activation energy and pre-exponential factor. The catalyst deactivation constant was found in a similar manner. Some students drew the graphs with only a few data points, while other groups used many data points which gave more accurate results. Several experimental runs were needed for accuracy of results and detection of noise which is built into the simulation.

### 5.5.5 CONTROL STRATEGY

Having calculated the rate constants and the catalyst deactivation constant, students were shown how to operate the computer program in the dynamic mode. Results obtained for the interim report were used for this second phase of the project. Students were given a second assignment letter informing them that they had been selected to act as consultants during the start up of a resid hydrotreater at Amoco's Texas City refinery. They were asked to simulate the start up of a single reactor, controlling the operation conditions manually to reach steady state. Then they ceased controlling the system and noted the time elapsed before automatic shut down.

The operating conditions within the reactor could only be controlled indirectly, by setting the temperature, flow rate, and composition of the feed stream. In addition, instantaneous changes in these variables were not permitted; students had to wait fifteen minutes between changes in the feed settings, and they were limited to changing the feed settings by no more than 100°F at a time. Finding a control strategy was typically a trial and error process. (Dynamic simulations were not charged against students' budgets).

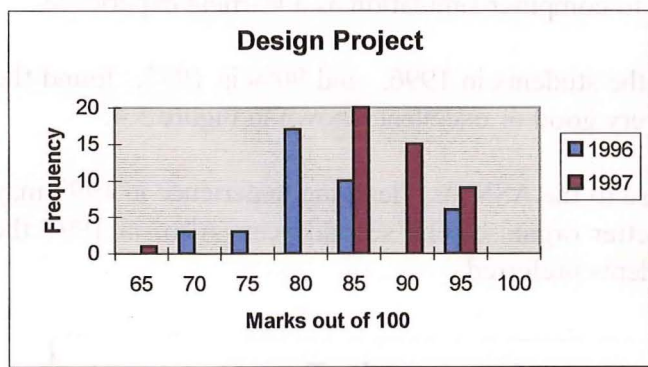
Two laboratory periods were allocated for determining the control strategy. Students then had a week to produce a full written report, including the results of their steady state experiments and an outline of their recommended start up procedure. Student reports were assessed for planning, decision making and problem solving.

## 5.6 EVALUATION

Student evaluations were an important part of the research for this thesis. Students evaluated the project by answering a questionnaire at the end of the course. As the project was undertaken in groups of three, students were also questioned on their experience of working in groups.

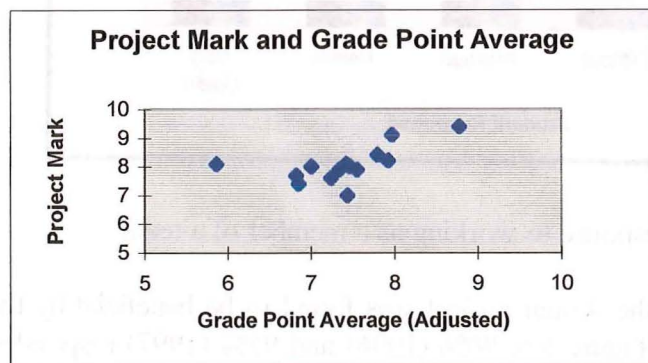
## 5.7 RESULTS AND DISCUSSION

The Amoco Project, worth 40% of the total marks for this course (roughly 10% of the years work), was carried out in groups of three, each member of the group receiving the same mark. In 1996, the mean of 80.6 out of 100 was high, and the standard deviation of 6.185 was low, indicating that all groups worked hard and achieved a high standard of work. Students in 1997 achieved a higher mean of 86.42 and an even lower standard deviation of 5.10, showing that there was not much variance in the high standard of projects.



Many of the students found the design project to be a challenge, and their efforts were suitably rewarded as indicated by their marks in Figure 5.2.

**Figure 5.2** Frequency distribution of students' marks for the design project in 1996 and 1997.



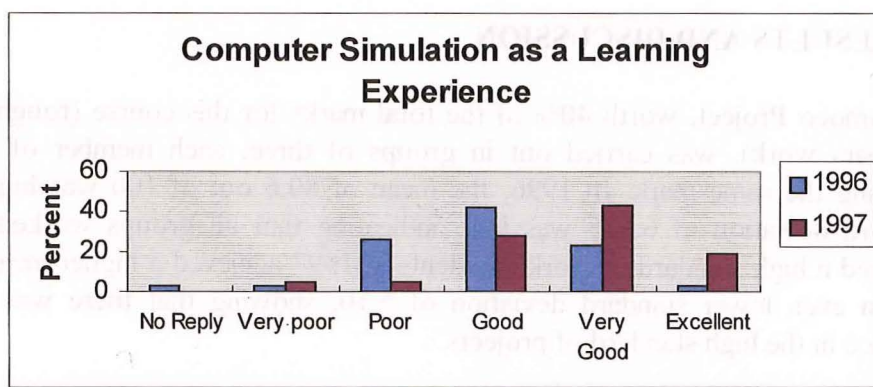
The assessment mark for each group for the Amoco project was compared with the group adjusted GPA for the 1996 examinations, Figure 5.3.

**Figure 5.3** Group project marks compared with adjusted GPA (1996).



The correlation in Figure 5.3 was not significant ( $r^2 = 0.58$ ) but the results were better than the comparison made with GPA and the problem solving assignment as shown in the previous chapter.

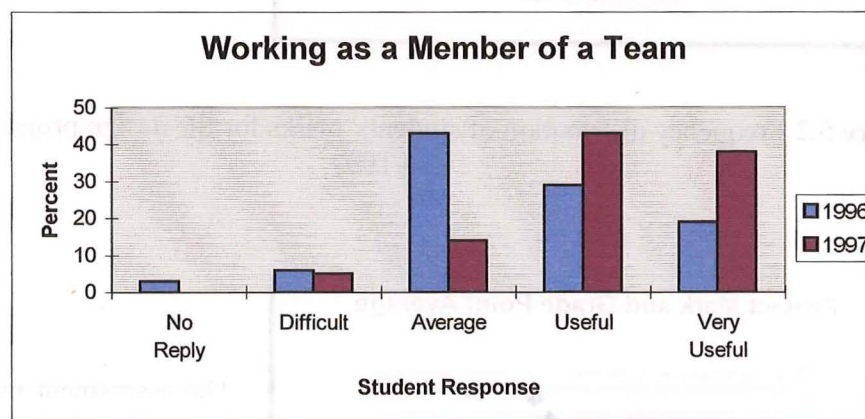
Figures 5.3, and 5.4 show data collected from questionnaires administered at the end of the design course for 1996 and 1997.



**Figure 5.4** Students' response to computer simulation as a learning experience.

As a learning experience, 68% of the students in 1996, and 90% in 1997, found the computer simulation to be good, very good or excellent; shown in Figure 5.4.

A reason for an increase in response to the ASM as a learning experience in 1997 may have been that the project was better organised the second year. Also, in 1997 the ASM was in colour which the students preferred.



**Figure 5.5** Students' response to working as a member of a team.

Working together as a team for the design project was found to be beneficial by the majority of students as shown in Figure 5.5; 90% (1996) and 95% (1997) responded with average, useful or very useful. Sharing ideas and discussing them were considered to be beneficial in the design project.



Students' comments on the advantages and disadvantages of working in groups are shown in Table 5.3.

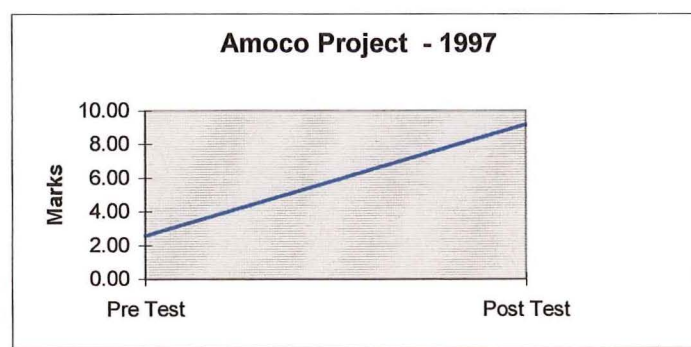
**Table 5.3** Advantages and disadvantages of working in groups.

Advantages	Disadvantages
Easier to solve problems in groups	Difficult to juggle time to work together
Reduced Workload	Personality clashes
Enhanced ability to find mistakes	Uneven sharing of the load
Sharing of ideas is valuable	
Talking through ideas is useful	

Time management is often a problem for students especially when they are involved in different laboratory groups, but they have to find time to work together for this project. The project has links with other subjects - reaction kinetics, reactor design, and data analysis covered in the second professional year.

## 5.8 LEARNING OUTCOME

In 1997, pre and post tests were carried out to determine the learning outcome of the Amoco design project. Figure 5.6 shows an upwards trend of mean scores (out of 10); 2.57 on the pre-test and 9.17 on the post-test indicating a significant learning outcome.



**Figure 5.6** Mean scores on the Amoco project pre and post tests.

Selected comments from 1997 students after completing the design project are in Table 5.4.

**Table 5.4.** Students' comments about the Amoco Project.

"Great fun, real life simulator"
"Ask the staff for help - only \$500 out of \$150,000 budget, why bother to look any further?"
"The free lab runs allowed on the first day were invaluable - we could try all sorts of things to get a feel for what we were trying to find, without worrying about paying for it".
"Its good to be able to do a project in class time".
"For me it has made it more real, more hands on. I feel like I understand the process better as a consequence".
"Excellent having lab times each week".
"The simulation was great, because it was interesting".

As the final assignment for our third year chemical engineering design class, the Amoco project provided a challenge for our students. In addition, the project provided an excellent practical problem solving exercise to conclude the design course.

### **6.1 COMPUTER-AIDED LEARNING**

Computer aided learning (CAL) was first introduced over two decades ago. In 1960 the University of Illinois launched the PLATO (Programmed Logic for Automatic Teaching Operations) project with the goal of designing a large computer-based system for instruction. CAL was not widely accepted at that time because it was regarded as a potential threat to teachers.

In the late 1970's, microcomputers were introduced into academic institutions and attitudes towards CAL then suddenly changed. It became possible for universities, schools and even individuals to own a computer which could be used for educational purposes. In the 1980's much of the CAL was drill and practice and the enthusiasm for CAL waned. Recently in the 1990's, authoring packages, such as Authorware 3.5, Quest, and Macromedia Director, have become available which have simplified the task of developing teaching and training material, as well as providing the ability to develop an interactive learning environment through the use of multimedia.

Computers have influenced the teaching and learning process of mathematics and science in many parts of the world. Research related to the use of mathematical packages has been mentioned in Chapter 3. Yong (1989) describes his experiences of successfully using CAL, tutoring and peer tutoring for teaching mathematics and science in Singapore. In a recent study (Williams and Zahed, 1996), the effectiveness of computer based training was compared with the standard or traditional lecture method. Results indicated that both groups demonstrated significant learning following training, and there was no significant difference in level of learning between the two groups.

This chapter discusses the analysis, design and development, implementation and evaluation of a computer module for teaching and learning finite difference approximations to first and second derivatives.

### **6.2 RESEARCH AIMS AND OBJECTIVES**

The research aims and objectives were:

- To promote active learning using a computer-based module on finite difference approximation to first and second derivatives.
- To enhance undergraduate Chemical Engineering education by providing alternative teaching tools that facilitate different learning styles.

- To study multimedia as an educational tool, and to evaluate the strengths and weaknesses of this technology in an educational setting.

### 6.3 PRODUCTION OF THE FDA COMPUTER MODULE

The production of a multimedia package required careful planning of the various stages as it was a complex process. In designing the Finite Difference Approximation (FDA) computer module, consideration was given to:

- content
- structure
- layout of screens
- the interface
- the overall design
- style of graphics

#### 6.3.1 Prototype and Development Process

Authorware 3.5 is a tool for developing an interactive computer application using multimedia which can be used by educators, corporate trainers and independent developers without the need to learn a complicated programming language. Authorware was specifically designed by instructional designers to create interactive learning, or computer-based training, pieces (Zielinski, 1996). In his book on Authorware, Zielinski describes the traditional development process for creating interactive programs (Figure 6.1).

The traditional approach to the development process is that of:



**Figure 6.1** The traditional five step development process.

The first phase in the creation of an interactive program has traditionally been the **analysis** phase. Analysis begins in order to determine how the needs of the end user can be met by the program. Every aspect of the program - from the statement of the problem to the type of hardware needed to run the interactive program - is explored in the analysis phase.

With a clear understanding of the objectives of the program and the end user's needs, the project moves into the second phase of creation known as **design**. In this phase, the scope of the project is defined. The first step is identification of the project content based on the issues that were identified in the analysis phase. The next step in the design phase is to create scripts and storyboards that will be used to actually develop the piece. Every screen is identified, then details of content are documented.

The central phase in the creation of an interactive program is *development*. The blueprint created in the design phase is converted into code according to the specifications. Once development begins, it is not uncommon for everyone to have new ideas which may cause the design to drift away from the agreed upon direction. Changes are inevitable and the process of managing these changes is a critical part of the development phase.

**Implementation** is the second to last phase when the project is delivered to the end user. This phase is also designed to ensure that the user is able to use the application and that it works in the target environment.

The phase that is commonly forgotten, is *evaluation*. In this phase, the success of the application is determined based on four specific criteria; reaction, retention, response, and return on investment.

The **reaction** of the end user has traditionally been collected through questionnaires immediately following the use of the program. Information gleaned about the application depended on questions asked in the questionnaires; what the user liked or did not like about their experience with the interactive software.

**Retention** can be measured in a scientific manner using a pre and post-test to judge the effectiveness of a piece. Issues surrounding long-term retention and mastery were not addressed, as the follow-up test needs to be conducted after a long period of time.

**Response** - The success of an application should be measured by the amount of behavioural change in the end user. During the analysis phase in this work, a set of objectives and the need for the application were defined.

**Return on Investment** - The more common measure of success for an interactive learning piece is the return on investment. The success of a piece of software is directly proportional to the amount of money that is saved over alternative methods of instruction, given that the same outcomes are achieved.

### 6.3.2 Disadvantages of the Traditional Development Methodology

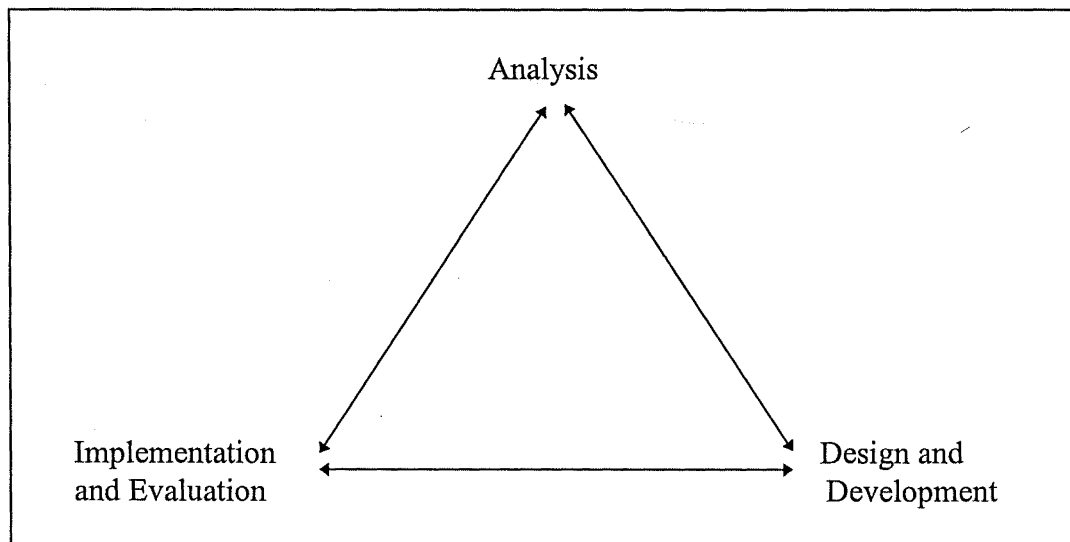
Over the last few years, developers of interactive learning programs have found that the traditional development approach is not the best approach to take because of several inefficiencies. There are three main disadvantages of this approach which may be summarised as:

- the gap in communication between the design and development teams,
- the fact that the best ideas arrive after development has begun,
- the end user is involved at the wrong end of the project.

### 6.3.3 A New Development Approach

A new approach to multimedia development which has attempted to take what was good from the traditional approach and combine it with the strengths of current technology to eliminate the three pitfalls has been created.

- Authorware is a tool that can be used in both the design and development phases. Design specifications, scripts and storyboards are replaced by prototypes that enable the developers and end users to see each other's ideas.
- As well as being easy to use, Authorware and the flow line structure are easy to change and maintain. As new ideas arise, these ideas can be incorporated with minimal effort.
- The new methodology was designed with the end user in mind. A piece can be built then immediately executed, allowing the end user to actually participate in the design and development efforts.



**Figure 6.2** Iterative phases of multimedia development.

This approach to multimedia development was the approach taken when designing and developing the Finite Difference Approximation application. All phases were combined in the development process; analysis, design and development, implementation and evaluation, as shown in Figure 6.2. The iterative process of changes and development are depicted by the central circles which start at the centre and work outwards towards delivery of the finished product.

At the analysis phase, instructional material and references were gathered to define the content. Characteristics of users, general aims and objectives, content structure, and types of interactivity were defined as part of the initial planning scheme.



What constitutes a good design? Designing and developing the multimedia package was the critical phase in the production process. This was an iterative process in which the design and development occurred as the package evolved. Simplicity was one of the key issues in the design as well as presentation of one or two key concepts on each screen. Other factors considered in the design were choosing the most appropriate content, and the best way of presentation, mindful of the user's needs, as well as creating an efficient and effective interface. Graphics, audio and motion sequences were created as the design developed. Manipulating data so that it was presented in a way that it could be easily understood by the user was a complex stage in development.

Authorware (Version 3.5) as the authoring tool, enabled links to be made with other software such as Microsoft Word - the Equation Editor, Excel and Coral Draw. After several months of design and development, the Finite Difference Approximations software was packaged into an executable file, copied to a CD ROM, ready for implementation and evaluation.

The third phase of multimedia development combined implementation with evaluation. It was necessary to get the opinions of the end users early in the process and through out development. This phase evaluated not only the end user's reaction to the application, but also checked to see that the product was teaching as intended. Production time was greater than 1000 hours for one hours teaching. Is the effort worthwhile?

#### 6.4 PILOT STUDY

A pilot study was conducted with 18 third year Chemical Engineering students. The study used a pre and post-test to compare the effects of two different learning methods as a supplement to lectures. Ten students studied a class handout on Finite Difference Approximations and eight students attended an optional laboratory session using the FDA computer module.

**Table 6.1** Results of pilot study pre-test

	<b>Handout</b>	<b>Module</b>
Mean	3.30	4.75
Std Dev	1.95	2.38
Number	10	8

Table 6.1 shows results of the pre-test, indicating that the group who did the module (Mean =4.75) had a better background knowledge of the subject than the group that worked on the handout (Mean =3.30). Another factor, was that the mean GPA of the students who did the module was 5.77 compared with 5.10 for the other group of students.

**Table 6.2.** Results of pilot study scores.

	<b>Handout</b>	<b>Module</b>
Mean	4.35	7.25
Std Dev	1.56	3.20
Number	10	8

Table 6.2 shows Mean and Standard Deviation for the difference scores of pre and post tests for the Pilot Study. Students who did the module achieved better scores (Mean = 7.25) than those who studied the handout (Mean = 4.35). A t-test for comparison of the difference of means resulted in  $t = 3.81$ ; a significant result at the 0.05 level of significance. As this was only a small sample of 18 out of a class of 45, further research needs to be done with a larger sample to confirm the significance of these results.

## 6.5 EXPERIMENTAL STUDY

An experimental study was conducted with 80 third year Mechanical Engineering students. A pre-test was given at the start of the first lecture on finite difference approximation, part of a course on numerical methods. Students were randomly divided into two groups for a one hour laboratory class; one group to work on a handout and the other the FDA computer module.

**Table 6.3.** Mean GPA for the two groups - Handout versus Module.

	<b>Handout</b>	<b>Module</b>
Mean	4.25	4.53
Std. Dev	1.52	1.46
Number	35	36

The Mean Grade Point Average (1996) of the two groups was compared to ensure that the randomly selected groups were of similar academic ability. Table 6.3 shows that there was very little difference between the two groups. The group who did the module had a Mean Grade Point Average of 4.53 compared with the group who did the handout, (Mean Grade Point Average was 4.25).

The group working on the handout was supervised by the class Lecturer and a Postgraduate student in the Mechanical Drawing Office, while the other group was supervised by the researcher and a Postgraduate student in the Glade CAD suite. Ideally, the lecturer would have supervised both sessions but this was not possible because of timetabling.

All students sat a test, a week after the tutorial session, contributing 5% towards their final assessment for the course. Following the test all students completed Solomon's Learning Styles Inventory for the researcher.



### 6.5.1 Handout Versus Module

The handout used for the Pilot study was rewritten by the researcher to ensure that the nomenclature was the same as that used by the Lecturer and the FDA computer module.

**Table 6.4.** Results of the differences in pre and post-test scores

	<b>Handout</b>	<b>Module</b>
Mean	11.65	9.94
Std Dev	4.71	4.48
Number	33	33

Table 6.4 shows that both groups demonstrated significant learning following teaching. Students who worked on the handout performed better than those who worked through the computer module. A t-test gave a result of 1.69 which was not significant at the 0.05 level of significance. An Analysis of Variance resulted in an F-test value of 2.29 which was also not significant at the 0.05 level of significance.

Students who worked on the computer module were confused by the method used for fitting a polynomial to unknown data points. In lectures students had been taught Newton's Difference Method (Griffiths and Smith, 1991) which is a numerical method compared with the algebraic method used in the computer module. Both methods were later discussed in a lecture the following day, and students were given the option of using one of the two methods to answer a question in the test. Despite their initial reaction to the algebraic method, about half the students chose to answer the test question using this method.

## 6.6 SOLOMAN'S LEARNING STYLES INVENTORY

As different learning methods were used to supplement lectures, students (n=75) completed a survey to determine their learning styles as defined by Soloman (1992).

Results of the Learning Styles survey were very similar to those of Felder (1988) and Montgomery (1995):

- 64% of the students learn best actively, yet many lectures, especially those to large classes, are typically passive;
- 56% of the students are sensors, yet we teach them intuitively;
- 89% of the students are visual, yet many lectures are verbal;
- 40% of the students are global; yet we seldom focus on the "big picture".

**Table 6.5** Results of the Learning Styles survey.

<b>Processing</b>		<b>Perception</b>		<b>Input</b>		<b>Understanding</b>	
Active	64%	Sensory	56%	Visual	89%	Sequential	60%
Reflective	36%	Intuitive	44%	Verbal	11%	Global	40%

Computer presentations (for example, Powerpoint<sup>TM</sup>) and coloured overheads provide a more visual aspect to University lectures. Many modern lecturers take advantage of these tools in their teaching.

Students have different learning styles, therefore educators need to provide a variety of teaching and learning methods in engineering education.

The objective of this thesis was *to assess computer applications and innovative teaching methods in Chemical Engineering education.*

The key attributes and desirable features of the methodology were:

- evaluation
- learning styles
- keep it simple
- learning-centred design of the user interface.
- significant learning outcomes
- enthusiastic and contented students.

In reviewing the HCI literature on approaches to interface design, an attempt was made to bring the strands of HCI design into focus and to highlight the advantages and disadvantages of the various approaches. Four approaches were identified: craft, cognitive engineering, software engineering and technologists. An important feature of each of the four approaches to interface design is their perception of quality in interface design.

What constitutes a good interface design? The criteria for an effective interface design method is a complex problem. HCI research must focus on the design of the interface considering the interaction of people and technology. The new challenge for designers and educators is the implementation of learner-centred design

A comparison of the mathematical packages- Maple, Mathematica, MATLAB and Excel using six engineering problems evaluated these packages in terms of their suitability for undergraduate engineering education.

The best application in a particular instance depends heavily on the nature of the problem. Maple had the advantage of giving a symbolic analytical solution but did not have the numerical capabilities of Mathematica, MATLAB or Excel. Each of the 3M's dealt well with symbolic manipulation and graphics. Excel displayed the most flexible graphics with, for example, the capacity to easily rotate three-dimensional plots. The Mathematica notebook provided an excellent interactive feature for documentation, report writing and teaching. The advantages of a particular application are lost if extensive work by hand is required to express the problem appropriately for that application.

Engineering graduates need to be skilled in a spreadsheet such as Excel, a programming language, and at least one of the other mathematical packages. MATLAB has advantages for matrix manipulation and linear systems, especially if its extensive optional toolboxes are relevant. Equally, Mathematica has distinct advantages in its use of natural language, the 'notebook' feature and user interface.

Maple was the most difficult package to learn and program, but was useful for verification of mathematical analysis. The best tool depends on individual needs and time spent learning the applications will reap the benefits of these powerful mathematical tools. This will result in an improvement in the teaching and learning outcomes of students during their degree course.

How to incorporate these powerful mathematical tools in our graduate and undergraduate courses is a key issue for engineering educators.

## **7.1 EVALUATION**

All teaching innovations introduced in this research were evaluated by student questionnaires, the results of which have been recorded and discussed in each chapter.

The development of problem solving skills is an integrated part of the teaching of design at the third year level of our Chemical and Process Engineering degree course. The computer modules were a useful supplement to lectures, providing an active learning environment for problem solving strategies. Working in pairs for problem solving was found to be beneficial by most of the students. Arranging a suitable time to work together was the main disadvantage.

Problem solving skills, at least for the small student populations used in this work, were found to be independent of academic achievement. Strategies for problem solving are techniques that can be learned. It is imperative that our graduates have the necessary skills and strategies to deal confidently with new situations and problems encountered in their professional careers.

Computer simulation and modelling of a chemical process can provide students with a different learning environment where they can investigate a chemical process by gathering data, performing data analysis and validating their results on the pilot plant. Testing their control strategy on the pilot plant simulation by changing the values of variables and observing results in the form of graphs gave students the opportunity to investigate an industrial process.

Time and budgetary constraints made students think carefully about their actions and decisions while working in groups for the project. Students enjoyed working in groups and appreciated the support of peers, as this project was a complex industrial design problem. A motivating factor in solving the problem was the industrial flavour and the relationship to the real world.

As the final assignment for our third year chemical engineering design class, the Amoco project provided a challenge for our students. In addition, the project provided an excellent practical problem solving exercise to conclude the design course.

## 7.2 LEARNING STYLES

Designing and developing a computer teaching package is a very iterative process and the package needs to be evaluated by many different end users. In the pilot study the FDA computer module was found to be more suitable for less able students and direct entry students who may not have knowledge of Taylor's series from previous mathematics study. The Balloon example (Excel spreadsheet) and error analysis (Matlab file) were found to be more appealing to the more able students.

Results of the experimental study showed that an organised problems class, with a well written handout and example problems to calculate, could be a useful supplement to lectures for learning mathematical concepts. A disadvantage of the computer laboratory is that it can be noisy and distracting especially if the network system is not functioning very well. Initially, these students were not willing to accept a numerical method which was different from that taught in lectures, affecting the results of this study.

Students have difference learning styles so lecturers need to provide a variety of teaching and learning methods in Chemical Engineering education.

## 7.3 RECOMMENDATIONS FOR FURTHER RESEARCH

- Design and administer a pre-test / post-test to measure the learning outcome for the computer problem solving modules.
- Survey graduates to investigate whether they find the learnt problem solving techniques useful in the workforce.
- Repeat the laboratory session with third year chemical engineering students when teaching Finite Difference Approximations to compare the experimental conditions of Handout and Module
- Administer the Learning Styles Inventory to chemical engineering students and compare the results with results obtained in this thesis.
- Further research is needed to investigate the algebraic skills of our students and reasons for their dislike and misunderstanding of algebraic concepts.



- Abell, M.E.L., Braselton, J.P. (1994). *Mathematica by Example*. Boston, USA: AP Professional.
- Adams, J. L. (1974). *Conceptual Blockbusting: A Guide to Better Ideas*. San Francisco: W.H. Freeman and Co.
- Allen, R.M., Gilmour, I.A., Mackenzie, J.G. (1996). Creative Problem Solving *Proceedings of 24<sup>th</sup> Australian and New Zealand Chemical Engineering Conference and Exhibition*. Sydney, Australia. Vol 4, 165-170
- Allen, R.M. and Mackenzie, J.G. (1996). Mathematical Power Tools -Mathematica, MATLAB and Excel. *Proceedings of Australian Association for Engineering Education, 7<sup>th</sup> Annual Conference and Convention*. Melbourne, Australia, 10-13 December, 1995.
- Apple. *Human Interface Guidelines: The Apple Desktop Interface*. Reading, M A: Addison-Wesley, 1987.
- Barker, H.A. and Zhuang, M. (1997). Control system analysis using Mathematica and a graphical user interface. *Computing and Control Engineering Journal*. 64-69, April
- Barnard, P. J. (1988). Approximating modelling of cognitive activity with an expert system: a theory based strategy for developing an interactive design tool. *Computer Journal*. **31**, 5, 445-456
- Barnard, P. J. (1991). Bridging between basic theories and the artefacts of human-computer interaction. in Carroll, J. M. (Ed) *Designing Interaction: Psychology at the Human-Computer Interface*. Cambridge University Press.
- Bickley, W.G (1948). Finite Difference formulae for the square lattice. *Quarterly Journal Mechanical and Applied Mathematics*. 1, p35-42
- Bramley, R. (1997). MATLAB 5.0 *IEEE Computational Science & Engineering*. 4 (1) : 89-90, Jan-Mar
- Bromilow, M. (1997). Computer algebra and applied mathematics. *IEE Review*, p259-262. November.
- Brown, J.S., Collins, A and Duguid. P. (1989). Situated Cognition and the Culture of Learning. *Educational Researcher*. 32-42. January-February.

- Buxton, W., Lamb, M. R. et al (1983). Towards a comprehensive user interface management system. *Computer Graphics* 17, 3, 35-42.
- Card, S.K., Moran, T. P. and Newel, A.(1983). *The psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates.
- Carnahan, B. (1992). Award Lecture. Computing in Engineering Education. From There, To Here, To Where? Part 2. Education and the Future. *Chemical Engineering Education*. Winter, p52-56.
- Carroll, J.W., Kellogg, W.A. and Rosson, M. B. (1991).The task-artefact cycle in Carroll, J.M. *Designing Interaction: Psychology at the Human-Computer Interface*. New York: Cambridge University Press,
- Chin, J.P., Diehl, V.A. and Norman, K.L (1988). Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface. *Proceedings of CHI Conference on Human Factors in Computing Systems*. New York: ACM, 213-218
- Conrad, A. (1997). Mathematical Modelling Program Receives Major Upgrade. *Microwaves & RF*, March
- Coughanowr, D.R. and Koppel, L.B. (1965). *Process Systems Analysis and Control*. New York: McGraw-Hill,
- Damodaran, L., Ip, K. and Beck, M. (1988). Integrating Human Factors principles into structured design methodology: a case study in the U.K. Civil Service. in Bullinger, H.J. (Eds) *Information Technology for Organisational Systems*. Elsevier, 235 -241.
- De Bono, E. (1967). *The Use of Lateral Thinking*. Harmondsworth: Penguin.
- De Bono, E. (1968). *The Five-Day Course in Thinking*. Harmondsworth: Penguin.
- De Bono, E. (1969). *The Mechanism of Mind*. Harmondsworth: Penguin.
- De Bono, E. (1973). *PO: Beyond Yes and No*, Revised Edition, Harmondsworth: Penguin.
- Dorgan, J.R. and McKinnon, J.T. (1996). Mathematica in the ChE Curriculum. *Chemical Engineering Education*. 136-141, Spring
- Etter, D.M. (1993). *Engineering Problem Solving with MATLAB*. Englewood Cliffs, New Jersey: Prentice-Hall,
- Felder, R. M. (1988). Teaching and Learning Styles in Engineering Education. *Engineering Education*(April), 674-681.



- Fogler, H. S. L., LeBlanc, S.E., and Montgomery, S.M. (1995). *Strategies for Creative Problem Solving*. Ann Arbor: University of Michigan.
- Fogler, H. S., Montgomery, S.M., and Zipp. (1992). Interactive Computer Modules for Undergraduate Chemical Engineering Instruction. *Computer Applications in Engineering Education*. 1, September-October, 11-12.
- Foster, K. (1996). Major Math Updates. *IEEE Spectrum*. 33 (8), 64-66, August
- Foster, K.R. (1997). Matrices and Much, Much More - MATLAB. *IEEE Spectrum*. 34 (2) : 14, Feb
- Fox, L. (1962). *Numerical Solutions of Ordinary and Partial Differential Equations*. London: Pergamon Press.
- Fox, L. (1968). *Computing Methods for Scientists and Engineers*. London: Pergamon Press.
- Gantt, H. L. (1910). *Work, Wages and Profits*. New York.
- Gerald, C.F. and Wheatley, P.O. (1984). *Applied Numerical Methods*. Massachusetts: Addison-Wesley.
- Griffiths, D.V. and Smith, I.M. (1991). *Numerical Methods for Engineers. A Programming Approach*. London: Blackwell Scientific Publications.
- Hartson, H.R., Siochi, A. C., and Hix. D. (1990). The UAN: A User-Orientated Representation for Direct Manipulation Interface Designs. *ACM Transactions on Information Systems*. 8, 3, 181-203
- Hausner, A. (1971). *Analog and Analog/Hybrid Computer Programming*. Englewood Cliffs, NJ: Prentice-Hall.
- Heck, A. (1993). *Introduction to Maple*. New York: Springer-Verlag
- Higgins, J. S. et. al. (1967). Identifying and Solving Problems in Engineering Design. *Studies in Higher Education*, 14(2), 169.
- Hix, D., and Hartson, H.R. (1993). *Developing User Interfaces: Ensuring Usability Through Product and Process*. New York: John Wiley and Sons.
- Hutton, J and Hutton, J. (1995). The Maple Computer Algebra System - A Review. *Journal of Applied Econometrics*. 10 (3) : 329-337, Jul-Sep
- Jackson, M. A. (1983). *System Development*. Prentice-Hall.
- Johnson-Laird, P.N. (1983). *Mental Models*, Cambridge: Cambridge University Press.

- Johnson-Laird, P.N., and Watson, P.C. (Eds) (1977). *Thinking: Readings in Cognitive Science*. Cambridge: Cambridge University Press.
- Johnson, P. and Johnson, H. (1991). Knowledge analysis of tasks: task analysis and specification for human-computer interaction. in Downton, A. (Ed) *Engineering the Human-Computer Interface*. McGraw-Hill.
- Jones, J. (1983). Students' Assessment of Instruction as a Basis for Teaching Improvement and Promotions Decisions: A Case Study. *Higher Education Research and Development*. Vol 2 (1), pp 3-14.
- Kahney, H. (1986). *Problem Solving : A Cognitive Approach*, Milton Keynes: Open University Press.
- Kahaner, D, Moler, C and Nash, S. (1989). *Numerical Methods and Software*. Englewood Cliffs, New Jersey: Prentice-Hall
- Kepner, C. H. and Tregoe, B.R. (1981). *The Rational Manager*. Princeton, NJ: Princeton Research Press.
- Kieras, D. E. and Polson, P.G. (1985). An approach to the formal analysis of user complexity. *International Journal of Man-Machine Studies*. **22**, 365-394
- Ko, E.I. and Hayes, J.R. (1994). Teaching Awareness of Problem-Solving Skills to Engineering Freshmen. *Journal of Engineering Education*. October.
- Kolb, D.A. (1984). *Experiential Learning: Experience as the Source of Learning and Development*. Englewood Cliffs, N.J: Prentice-Hall.
- Lander, T.K. (1987). Relations between cognitive psychology and computer systems design. In Carroll, J (Ed) *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction* Cambridge, M A: MIT Press.
- Laurel, B. (1990). *The Art of Human-Computer Interface Design*, New York: Addison-Wesley.
- Laurillard, D. (1996). Rethinking University Teaching: A framework for the effective use of educational technology. London: Routledge.
- Lockyer, K. G. , Gordon, J. (1991). Critical Path Analysis and Other Project Network Techniques. London: Pitman.
- Macaulay, L. (1993). Co-Operative requirements capture *Unpublished lecture*.
- Maul, G.P., and Gillard, J.S. (1996). Teaching Problem Solving Skills. *Computers and Industrial Engineering*. 31 (1/2) 17-20.

- McCarthy, B. (1987). *The 4MAT System: Teaching to Learning Styles with Right/Left Mode Techniques*. Barrington, Illinois: Excel, Inc.
- Montgomery, S. M. (1995). *Addressing Diverse Learning Styles Through the Use of Multimedia*. Frontiers in Education Conference Proceedings.
- Moran, T.P. (1981). The Command Language Grammar. *International Journal Man-Machine Studies*. **15**, 3-50
- Morrison, D. (1993). *University of Minnesota Solar Vehicle Project*. Minnesota: University of Minnesota.
- Munro, N. (1997). Symbolic algebra computing in control engineering. *Computing and Control Engineering Journal*. 8, **2**, 50-76, April.
- Nakamura, S (1996). *Numerical Analysis and Graphical Visualisation with Matlab*. Englewood Cliffs, NJ: Prentice-Hall.
- Nelson, T.H. (1990). The right way to think about software design in Laurel, B. (Ed) *The Art of Human-Computer Interface Design*. New York: Addison-Wesley.
- Nelson, T.H. (1974). *Computer Lib and Dream Machines*. Chicago: Hugo's Book Service.
- Newell, A. and Simon, H.A. ( 1972). *Human Problem Solving*, Englewood Cliffs, NJ: Prentice-Hall.
- Noss, R. *Reading the Sines*. (1995). Report on Teaching and Learning with Technology Programme: Project 15. Imperial College, London and The University of Leeds
- Norman, D. A. (1988). *The Psychology of Everyday Things*. USA: Basic Books
- Norman, D.A. and Draper, S.W. (1986). *User Centred Design*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.
- Norman, D.A. and Spohrer, J.C. (1996). Learner-Centred Education. *Communications of the ACM*. April, Vol 39, No 4.
- Olsen, D.R. and Dempsey, E.P. (1983). SYNGRAPH: a graphical user interface generator. *Computer Graphics* **17**,3, 43-50
- Oren, T., Salomon, G., et al. (1990). Guides: characterising the Interface. in Laurel, B. (Ed) *The Art of Human-Computer Interface Design*. New York: Addison-Wesley.
- Orsak, G.C. and Etter, D.M. (1995). Using the Internet and MATLAB. *IEEE Signal Processing Magazine*. 12 (6): 23-32, Nov

- Orvis, W.J. (1996). *Excel for Scientists and Engineers*. (2<sup>nd</sup> Ed) San Francisco: Sybex.
- Ozisk, M. N. (1994). *Finite Difference Methods in Heat Transfer*. London: CRC Press.
- Park, K., and Travers, K.J. (1995). *A Comparative Study of a Computer-based and a Standard College Freshman Calculus Course*. Unpublished Report
- Pattee, H. A. (1995). Selecting computer mathematics. *Mechanical Engineering*. September
- Payne, S.J. and Greene, T. R. C. (1986). Task-Action Grammars: a model of the mental representation of task-languages. *Human-Computer Interaction* **2**, 93-133.
- Peet, N. J. (1991). *The use of input-output methods to evaluate the energy requirements and environmental consequences of economic activity. A Practical Introduction*. Chemical and Process Engineering, University of Canterbury,.
- Polya , G. (1945). *How to Solve it*, (2<sup>nd</sup> Ed., 1957). New York: Doubleday.
- Polya , G. (1962). *Mathematical Discovery, Vol 1: On Understanding, Learning and Teaching Problem Solving*, New York: John Wiley.
- Preece, J. (1994). (Ed) *Human-Computer Interaction*, New York: Addison-Wesley.
- Roache, P. (1972). *Computational Fluid Dynamics*. New Mexico: Hermosa Publishers.
- Schumucker, B. (1982). MacApp: an application framework. *Byte* **7**, 242-282
- Schaufelberger, W. (1997). The Student Edition of MATLAB. *Automatica*, **33** (5) 1005-1006.
- Seader, J. D. (1989). Education and Training in Chemical Engineering Related to the Use of Computers. *Computing and Chemical Engineering*. Vol 13, pp 377
- Seiter, C. (1992). Mathematics Analysis. *Macworld*, June
- Shneiderman, B (1992). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. New York: Addison-Wesley.
- Simon, T. (1988). Analysing the scope of cognitive models in human-computer interaction: a trade-off approach. in Jones, D.M. and Winder, R. (Ed) *People and Computers IV* Cambridge University Press, 79-93.

- Smith, D.C. Irby, C. et al (1982). Designing the Star User Interface. *Byte*. 7, 242-282
- Smith, G.D. (1965). *Numerical Solutions of Partial Differential Equations*. London: Oxford University Press.
- Snow, R. and Farr, M. (1987). *Vol 3: Cognitive and Affective Process Analysis*. Hillsdale, NJ: Erlbaum.
- Soloman, B. (1992). *Inventory of Learning Styles*. North Carolina State University. USA.
- Squires, R.G., Reklaitis, G.V., Yeh, N.C., Mosby, J.F., Karimi, I.A., and Andersen. P.K. (1991). Purdue-Industry Computer Simulation Modules -The Amoco Resid Hydrotreater Process. *Chemical Engineering Education*, 25 (2), 98-101, Spring.
- Studt, T. (1996). Mathematica 3.0 Adds Equation Editor for Presentation Graphics. *R&D Magazine*. 38 (8):33, July
- Sutcliffe, A. (1988). Some experiences in integrating specifications of human-computer interface design with structured systems development method. in Jones, D. M. and Winder, B. (Ed) *People and Computers IV*. Cambridge University Press, 146-160.
- Thorn, A and Apelt, C.J. (1961). *Field Computations in Engineering and Physics*. London: D Van Nostrand.
- Tognazzini, B. (1992). *Tog on Interface*. New York: Addison-Wesley.
- Url, J.J. (1995). Calculus and Mathematica. *U M E Trends*. 6, 6, p15
- Vaughn, S.M. (1997). Review of Mathematica. *Human and Ecological Risk Assessment*. 3 (3), 343-349
- Wallace, M.D. and Anderson, T.J. (1993). Approaches to Interface Design. *Interacting with Computers* 5, 3, 259-278.
- Walsh, P. (1989). Analysis for task object modelling (ATOM): towards a method of integrating task analysis with Jackson System Development for user interface software design. in Diaper, D (Ed) *Task Analysis for Human-Computer Interaction* Ellis Horwood, 186-209.
- Wassermann, A.J., and Pircher, P.A. (1987). Developing Interactive Information Systems with the User Software Engineering Methodology. In: Baecker, R.M. and Buxton, W.A.S. (Eds) *Readings in Human-Computer Interaction- A Multidisciplinary Approach*. Morgan-Kaufmann, 508-527.

- Waynant, R. and Fuller, R. (Eds) (1997). Software and the Web. *Circuits and Devices*, May.
- Wertheimer, M. (1961). *Productive Thinking*. London: Tavistock
- Whimbey, A. and Lochhead, J. (1980). *Problem Solving and Comprehension: A Short Course in Analytical Reasoning*. Philadelphia: Franklin Press.
- Whiteside, J., Bennett, J. and Holtzblatt, K. (1988). Usability engineering. In: Helander, M. (Ed) *Handbook of Human Computer Interaction*. Amsterdam : New Holland. 791-817.
- Wickelgren, W.A. (1938). *How to Solve Problems*. Revised Edition. (1974). San Francisco: Freeman.
- Williams, T.C. and Zahed, H. (1996). Computer-Based Training Versus Traditional Lecture - Effect On Learning and Retention. *Journal of Business and Psychology*, 11 (2) 297-310.
- Winogard, T and Flores, F (1986). *Understanding Computers and Cognition: A New Foundation for Design*. Norwood, New Jersey: Ablex.
- Wittrock, M.C. (1986). 'Students' thought processes', in M .C. Wittrock (Ed) *Handbook of Research in Teaching*, New York: Macmillan, 297-314
- Woods, D.R., Wright, J.D., Hoffman, T.W., Swartman, (1975). R.K. and Doig, I.D., Teaching Problem Solving Skills. *Engineering Education*. **55** (3), 238-243.
- Woods, D.R. (1985). *The McMaster Problem Solving (MPS) Program*. Department of Chemical Engineering, Ontario, Canada: McMaster University.
- Woods, D. R. (1983). *A Strategy for Problem Solving* (Course Notes, 3rd Ed. ).
- Wroblewski, D.A. (1991). The construction of human-computer interfaces considered as a craft. in Karat, J (Ed) *Taking Software Design Seriously*. Academic Press, 1-19
- Yong, Yeow-Chin. (1989). CAL for Mathematics and Science Teaching, in P. Barker, (Ed) *Multi-Media Computer-Assisted Learning*. NewYork: Kogan Page, 78-96.
- Young, B.R. and Svrcek, W.Y. (1996). The Application of Steady State and Dynamic Simulation for Process Control Design of a Distillation Column with a Side-Stripper. *Proceedings of 24<sup>th</sup> Australian and New Zealand Chemical Engineering Conference and Exhibition*. Sydney, Australia, 30 September - 2 October, 1996. Vol 2, p 145-150.

- Young, R.M., Green, T.R.G. and Simon, T. (1989). Programmable user models for predictive evaluation of interface designs. in Brice, K. and Lewis, C.H. *Proceedings of CHI'89: Human Factors in Computing Systems*. ACM, 15-19.
- Zielinski, R.S. (1996). *Special Edition. Using Macromedia Authorware 3.5*. Que Corporation. USA:
- Zienkiewicz, O.C. and Morgan, K (1983). *Finite Elements and Approximation*. New York: John Wiley and Sons.

The following publications have resulted from work for this thesis:

- Allen, R.M. and Mackenzie, J.G. *Computers - An educational tool in Engineering Education*. 6<sup>th</sup> Australasian Association of Engineering Education Conference. Sydney, Australia, 10-14 December, 1994.
- Allen, R.M. and Mackenzie, J.G. *Problem-based Learning in Engineering Computation*. IPENZ Annual Conference, Palmerston North, 10-14 February, 1995.
- Allen, R.M. and Mackenzie, J.G. *Computers as a Tool for Teaching Report Writing*. The Australian Society for Computers in Learning in Tertiary Education Conference, Melbourne, Australia, 4-6 December, 1995.
- Allen, R.M., Boustead, T. M., and Mackenzie, J.G. *Mathematical Power Tools - Maple, Mathematica, MATLAB versus Excel*. The Australian Association for Engineering Education 7<sup>th</sup> Annual Conference and Convention, Melbourne, Australia, 10-13 December, 1995.
- Allen, R.M., and Mackenzie, J.G. *Problem Solving with Mathematica, MATLAB and Excel*. IPENZ Annual Conference, Dunedin 9-13 February, 1996.
- Allen, R.M., Gilmour, I.A., and Mackenzie, J.G. *Computer-based Learning of Creative Problem Solving*. Post Graduate Science and Technology Conference. University of Canterbury, Christchurch, July, 1996.
- Allen, R.M., Gilmour, I. A., and Mackenzie, J.G. *Creative Problem Solving* 24<sup>th</sup> Australian and New Zealand Chemical Engineering Conference and Exhibition, Sydney, Australia, 30 September - 2 October, 1996
- Allen, R.M., Gilmour, I.A., and Mackenzie, J.G. *Teaching Problem Solving Skills in Engineering Design*. IPENZ Annual Conference, Wellington, 6-10 February, 1997
- Allen, R.M. Gilmour, I.A., and Mackenzie, J.G. *Computer Simulation in Chemical Engineering Education - the Amoco Resid Hydrotreater Process*. 25<sup>th</sup> Australian and New Zealand Chemical Engineering Conference and Exhibition, Rotorua, New Zealand, 28 September - 1 October, 1997.
- Mackenzie, J.G. and Allen, R.M. *Mathematical Power Tools - Maple, Mathematica, MATLAB and Excel*. *Chemical Engineering Education* - Accepted for publication 15/12/97.



The handout on finite difference approximations to first and second derivatives given to students to compare two different teaching methods, as described in Chapter 6, was as follows:

## ***1.0 First Derivative Approximations***

### ***1.1 First Order***

Recall the definition of the derivative with respect to  $x$  of the function  $f(x)$  at  $x = x_0$ :

$$\frac{df}{dx} = \lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h} \quad (1)$$

If we have  $f(x)$  available at discrete points

$$x_{i+1} = x_i + h$$

where  $i$  is an integer, then we can use the Taylor's series expansion of  $f(x)$  to derive difference approximations. Taylor's series not only derives the difference formulas systematically, but also the error terms.

The Taylor's series expansions of a function  $f(x)$  about a point  $x_i$  is:

$$f_{i+1} = f_i + hf'_i + \frac{h^2}{2!} f''_i + \frac{h^3}{3!} f'''_i + \frac{h^4}{4!} f^{(4)}_i + \dots \quad (2)$$

To solve for  $f'_i$ , we can rearranged equation (2):

$$hf'_i = f_{i+1} - f_i - \frac{h^2}{2!} f''_i - \frac{h^3}{3!} f'''_i - \dots \quad (3)$$

$$f'_i = \frac{f_{i+1} - f_i}{h} - \frac{h}{2} f''_i - \frac{h^2}{6} f'''_i - \dots \quad (4)$$

$$f'_i = \frac{f_{i+1} - f_i}{h} + \varepsilon \quad (5)$$

where the error is:

$$\varepsilon \approx -\frac{h}{2} f_i'' \quad (6)$$

Equation (5) is a **forward** difference approximation to the derivative at  $x_i$  because it is obtained by stepping in the forward or increasing direction in  $x_i$

This approximation to the derivative is said to be **first order** correct because the **truncation error** is proportional to the step size  $h$  to the first power.

**Truncation error** refers to the error arising from truncating the Taylor's series expansion to a finite number of terms. It should not be confused with round-off error arising from the limited number of bits representing numbers in a digital computer.

The **backward** difference approximation for the first derivative using  $f_{i-1}$  and  $f_i$  is obtained in a similar manner:

$$f_i' = \frac{f_i - f_{i-1}}{h} + \varepsilon \quad (7)$$

where the error is:

$$\varepsilon \approx \frac{h}{2} f_i'' \quad (8)$$

which is also **first** order correct.

## 1.2 Second Order

The **central** difference approximation using  $f_{i+1}$  and  $f_{i-1}$  may be derived by Taylor's expansion:

$$f_{i+1} - f_{i-1} = 2hf_i' + \frac{1}{3}h^3 f_i''' + \dots \quad (9)$$

where the  $f_i''$  term has been automatically eliminated. Solving for  $f_i'$  we get:

$$f_i' = \frac{f_{i+1} - f_{i-1}}{2h} - \frac{1}{6}h^2 f_i''' + \dots \quad (10)$$

Including the error term, the **central** difference approximation is expressed as:

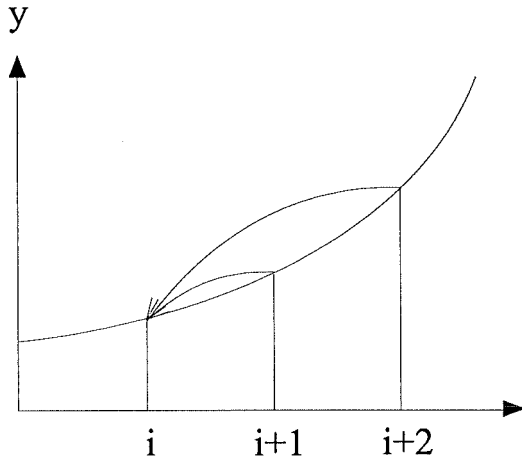
$$f_i' = \frac{f_{i+1} - f_{i-1}}{2h} + \varepsilon \quad (11)$$

where

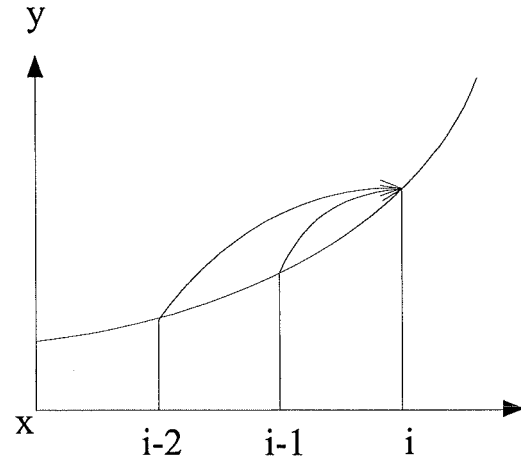
$$\varepsilon \approx -\frac{h^2}{6} f_i''' \quad (12)$$

Equation (11) is a **central** difference approximation because it uses values from steps each side of  $x_i$  to approximate the derivative at  $x_i$ . The error of the central difference approximation is proportional to  $h^2$ ; that is **second** order.

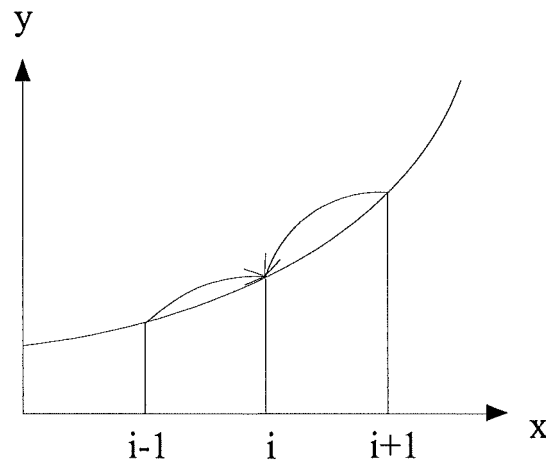
As the number of data points increases, a more accurate difference approximation can be obtained. Let's look at three points.



**Figure A2.1** Forward Difference Approximation



**Figure A2.2** Backward Difference Approximation



**Figure A2.3** Central Difference Approximation

Let's derive a difference approximation for  $f'_i$  using three points  $f_i, f_{i+1}, f_{i+2}$

$$f_{i+1} = f_i + hf'_i + \frac{h^2}{2!} f''_i + \frac{h^3}{3!} f'''_i + \frac{h^4}{4!} f^{(4)}_i + \dots \quad (13)$$

$$f_{i+2} = f_i + 2hf'_i + \frac{(2h)^2}{2!} f''_i + \frac{(2h)^3}{3!} f'''_i + \frac{(2h)^4}{4!} f^{(4)}_i + \dots \quad (14)$$

If we multiply equation (13) by 4, the second derivative term is cancelled when we subtract equation (14), resulting in the third derivative term as the leading error term.

Subtracting equation (14) from 4 times equation (13) we get:

$$4f_{i+1} - f_{i+2} = 3f_i + 2hf'_i + \frac{2h^3}{3!}f_i''' + \dots \quad (15)$$

Solving for  $f'_i$  we get

$$f'_i = \frac{-f_{i+2} + 4f_{i+1} - 3f_i}{2h} + \varepsilon \quad (16)$$

where the error is given by:

$$\varepsilon \approx \frac{h^2}{3}f_i''' \quad (17)$$

Equation (16) is the three point **forward** difference approximation, which is **second** order correct; that is,  $h$  in the error term (equation 17) is of the power degree two.

Similarly, the three point backward difference approximation may be derived using  $f_i, f_{i-1}, f_{i-2}$ :

$$f'_i = \frac{3f_i - 4f_{i-1} + f_{i-2}}{2h} + \varepsilon \quad (18)$$

where the error is:

$$\varepsilon \approx \frac{h^2}{3}f_i''' \quad (19)$$

## 2.0 Second derivative approximation

### Second Order

Adding equations (20) and (21) results in the first and third derivative terms in the Taylor's series cancelling out.

$$f_{i+1} = f_i + hf'_i + \frac{h^2}{2!}f_i'' + \frac{h^3}{3!}f_i''' + \frac{h^4}{4!}f_i'''' + \dots \quad (20)$$

$$f_{i-1} = f_i - hf'_i + \frac{h^2}{2!}f_i'' - \frac{h^3}{3!}f_i''' + \frac{h^4}{4!}f_i'''' - \dots \quad (21)$$

The result rearranges to:

$$f_i'' = \frac{f_{i-1} - 2f_i + f_{i+1}}{h^2} \quad (22)$$

which is a **second** order correct **central** difference approximation.

The truncation error is:

$$\varepsilon \approx -\frac{h^2}{12} f_i'''' + \dots \quad (23)$$

To obtain a **second** order correct **forward** difference approximation to the second derivative, a four point formula is needed.

$$f_i'' = \frac{2f_i - 5f_{i+1} + 4f_{i+2} - f_{i+3}}{h^2} + \varepsilon \quad (24)$$

where the error is:

$$\varepsilon \approx \frac{11}{12} h^2 f_i'''' \quad (25)$$

Similarly, the backward difference approximation :

$$f_i'' = \frac{-f_{i-3} + 4f_{i-2} - 5f_{i-1} + 2f_i}{h^2} \quad (26)$$

where the error is:

$$\varepsilon \approx \frac{11}{12} h^2 f_i'''' \quad (27)$$

### 3.0 First Derivative Approximation Third Order

The four point approximation in equation (28) is **third** order correct and so more accurate again.

$$f_i' = \frac{-11f_i + 18f_{i+1} - 9f_{i+2} + 2f_{i+3}}{6h} + \varepsilon \quad (28)$$

where the error is:

$$\varepsilon \approx -\frac{h^3}{4} f_i'''' \quad (29)$$

Note that for any derivative approximation, the sum of the coefficients in the approximation must be zero. For example, in equation (28), the coefficients sum to  $\dots 11+18-9+2=0$ .

## References

- Nakamura, S. (1996) Numerical Analysis and Graphics Visualisation with Matlab. Englewood Cliffs, N J: Prentice-Hall.
- Griffiths, D.V. and Smith, I.M. (1991) Numerical Methods for Engineers. A Programming Approach. London: Blackwell Scientific Publications.

## Exercises

1. Given the function  $f(x) = x^3$ ,  $x=5$ ,  $h=1$  calculate the difference approximations using equations 5, 7, 11 and 16, and the corresponding errors using equations 6, 8, 12, and 17. Compare the results with the exact (analytical) values.
2. Given the function  $f(x) = x^5$ ,  $x=5$ ,  $h=1$  calculate the difference approximations using equations 22, 24, and the corresponding errors using equations 23 and 25. Repeat the calculations for  $h=0.5$ . Compare the results with the exact (analytical) values.
3. Given the function  $f(x) = x^4$ ,  $x=1$ ,  $h=1$  calculate the difference approximation using equation 28. Calculate the errors using equations 29. Repeat the calculations for  $x=10$ . How well do the results agree with the analytical values?
4. Calculate the acceleration and velocity of a weather balloon using the data and formulae given in the spreadsheet. Use approximations to first derivative for velocity and approximations to second derivative for acceleration.